6 SYSTEM/VALUE ENGINEERING

6.1 INTRODUCTION

The systems engineering process is a proven disciplined approach that supports management in clearly defining the mission or problem; managing system functions and requirements; identifying and managing risk; establishing bases for informed decision making; and verifying that products and services meet customer needs. An overview of the process is shown in Figure 6-1 below.

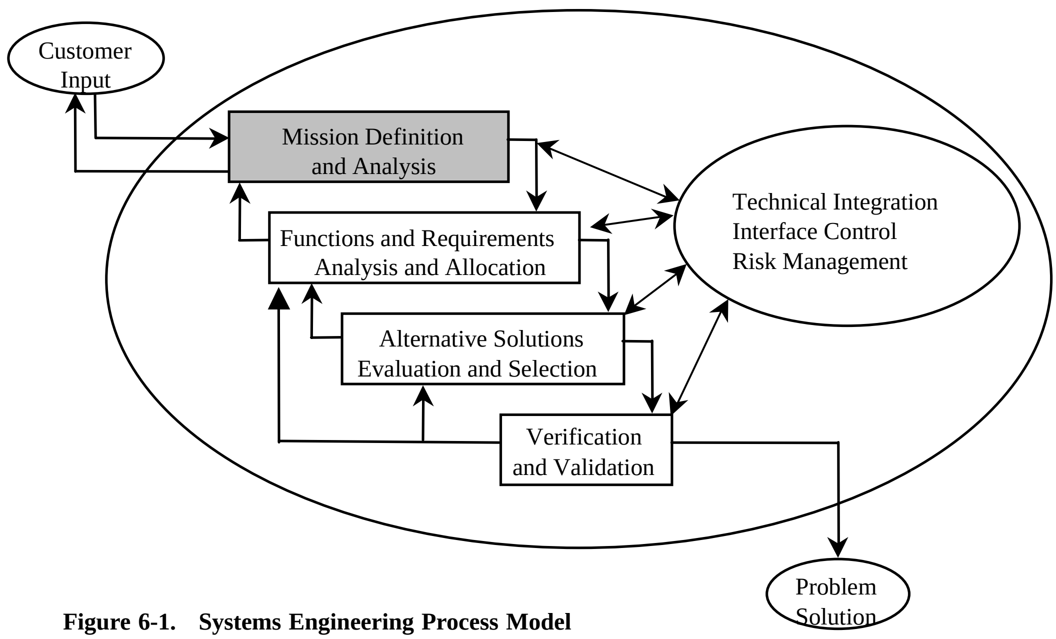


Figure 6-1.   Systems Engineering Process Model

6.2 PURPOSE

The purpose of this systems engineering methodology process description is to identify the steps of the systems engineering process and to provide implementation guidance by presenting recommended proven techniques and methods that may be used for accomplishment of selected process steps. Specific techniques and methodologies used in implementation of the systems engineering process, describing and recommending acceptable “HOW TO’s” for these steps are provided in this section of the manual. They are intended for application where specific methods are not covered by existing orders or other site-specific implementation tools.

6.3 Section Structure

This section is structured to describe the recommended methodologies and techniques in self-contained appendices. This structure accommodates both additions and revisions to these appendices as appropriate. The appendices are as follows:

Appendix A: Mission Definition

This appendix, with attachments, describes the steps and techniques to be used for mission definition in the application of the systems engineering process. The intent is to provide the user with guidance in working with their customers to translate stated needs and objectives into a concise and defendable definition of the work to be performed. The use of this guide will assist the user in developing the first step in the systems engineering process, i.e., clearly defining the problem and the customer's need.

Appendix B: Function and Performance Requirements Development

This appendix describes a process for the development of functions and performance requirements. Two methods for functional development are presented along with a discussion of performance requirements development and key attributes of good requirements. Example functional hierarchy diagrams, functional flow block diagrams, N-squared diagrams, and enhanced functional flow block diagrams are provided.

Appendix C: Alternative Studies and Value Engineering

This section, with attachments, describes the steps, tools, and techniques involved in performing Alternative Studies for selecting the optimum, most cost-effective, alternative that meets an activity's functions and requirements. Value engineering studies, which are a specific type of alternative study, are included.

D. Interface Control - To Be Developed

This section will describe the steps and techniques to be used for Interface Control in the application of the systems engineering process. The intent is to provide the user with guidance on how to identify and control system interfaces. Examples of how to document interface requirements are provided.

E. Systems Engineering Management Plan (SEMP) - To Be Developed

This section will describe the process involved in developing a system engineering management plan (SEMP) for a program, project or engineering task. This guide is written to be used in conjunction with the other sections in this manual. Guidance is provided on when a SEMP is needed and the recommended content of a SEMP.

Appendix A

MISSION DEFINITION

A.1.0 Introduction

Mission Definition establishes a solid foundation for proceeding with a work task by understanding, confirming, and documenting the change or problem being addressed and the criteria for success. Mission Definition is the initial activity performed in the application of the systems engineering process to define what must be done to satisfy the customer's need.

A.1.1 Purpose

The purpose of this guide is to describe the steps and techniques to be used for Mission Definition in the application of the systems engineering process. The intent is to provide the user with guidance in working with their customers to translate stated needs and objectives into a concise and defendable definition of the work to be performed.

The use of this guide will assist the user in developing the first step in the systems engineering process, i.e., clearly defining the problem and the customer's need. When properly performed, the Mission Definition step will answer the questions:

- What are trying to do (problem)?
- Why are we doing this (basis)?
- What is the initial state (present condition)?
- What are the boundaries (limits)?
- What is the outcome we seek (goals/objectives)?
- What is the final state (desired outcome)?
- How do we measure progress or achievements (success criteria)?

This guide will focus on the need to develop and document a concise definition of the problem, a firm basis and rationale for the work, the boundaries for the task, the customer requirements to be satisfied, and the goals and objectives to be achieved.

A.1.2 What is Mission Definition?

The key to the successful execution of a project or task, Mission Definition is the concise definition of the work to be performed with a clear understanding of the expected outcome. It is the translation of the customer's stated needs and objectives into the definitive set of the highest level function(s) and performance requirements necessary to accomplish the task, including the rationale and justification for each.

In this context, the term "mission" should be taken as the highest level function(s) to be performed by the task, i.e., what has to be done to change the initial state (current condition) to the final state (desired outcome). Mission Definition includes clear and concise problem and mission statements, the drivers that result in the need for the proposed activity, the highest level performance requirements associated with the major function(s), high level external interfaces, and identification of risks. Mission Definition may also include the identification of the highest level systems to be developed or modified by the task and/or proposed alternatives for consideration, as appropriate.

A.1.3 When Should Mission Definition be Performed?

Mission Definition is performed at the initiation of work with the customer and is the start of the systems engineering process, as shown in Figure 1. It serves as a "contract" with the customer to define, establish boundaries for, and document the scope and expectations of the task. A graded Mission Definition should be performed at the start of all tasks, regardless of complexity, to assure the work to be performed is precisely specified and understood. Whether the complexity of the task demands the use of software tools (e.g., CORE®) to capture the information, or is sufficiently simple to be "done in your head", the intent is fundamentally the same; develop, document, and agree to a complete, clear, and technically accurate definition of the work the customer needs to have performed.

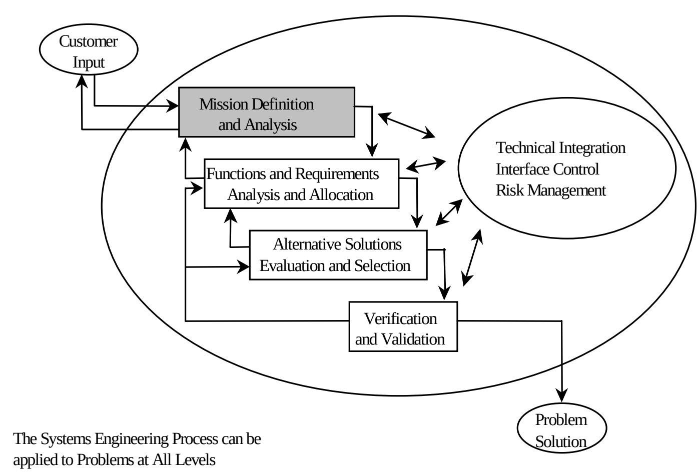


Figure A1. Systems Engineering Process Model

A.2.0 Methodology

The method used to perform the Mission Analysis, discussed below, is illustrated in the diagram shown in Figure A2.

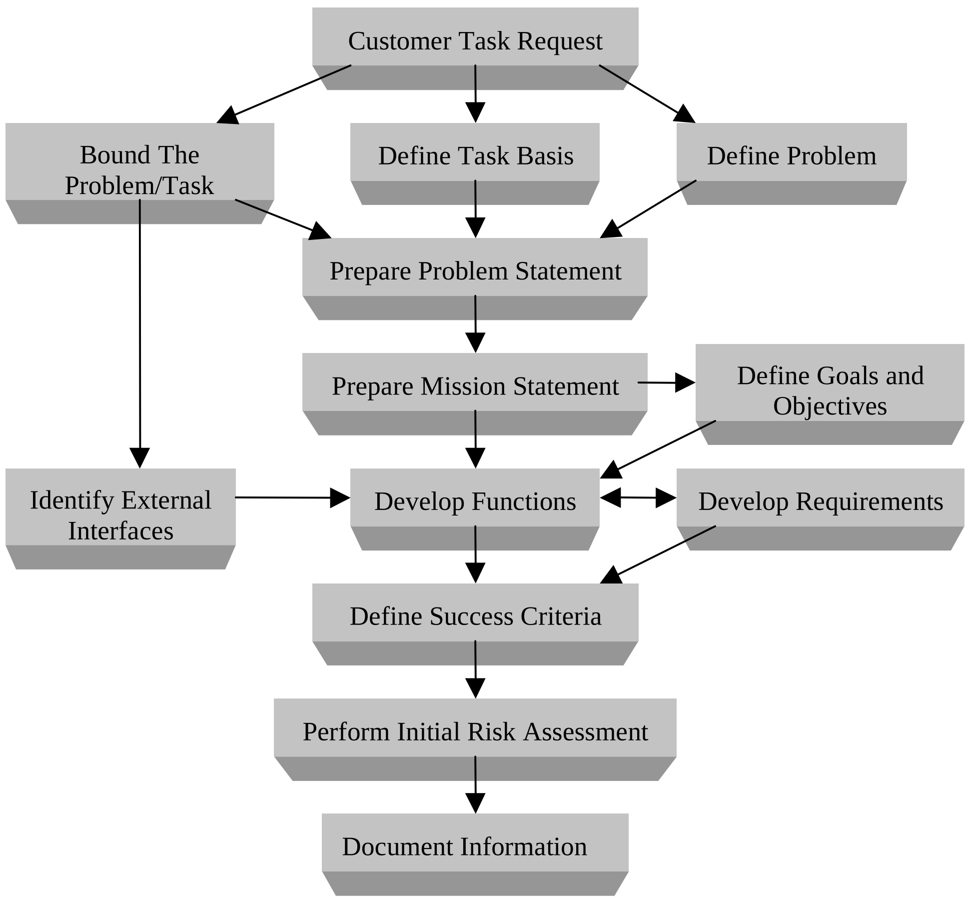


Figure A2. Mission Definition Methodology

A.2.1 Customer Task Request

Work is initiated following the receipt of a written or verbal request from a customer to perform a task. Ideally, the task request should identify the problem to be solved or corrected, the goals and objectives to be achieved, and the criteria for success. Often, however, this work request is incomplete, has no discernable basis or rationale, and/or worse, is a command to implement a preconceived “solution” to an undefined problem.

The customer task request should be used to initiate a probing discussion with the customer and their technical experts to begin to develop a precise and clear definition of the work to be performed. It is essential that all information obtained through these discussions is thoroughly documented.

A.2.2 Problem Definition and Customer Needs

This step of the process concentrates on clearly understanding and defining the problem and customer needs before proceeding with the task. Too many tasks are conducted without a clear understanding of what needs to be done. This leads to rework or possible failure.

A.2.2.11 Basis for the Task Request

To better define the task request, it is helpful to initially understand the basis and rationale for why the task is necessary. Question the customer on what the drivers are that make the requested work needed; ask why the customer needs to have the work done, for what purpose, and for whom.

To illustrate, consider the following example. Assume the customer’s task request is to “Upgrade the Q-Lab Facility”. By probing and asking questions as to why the upgrade is necessary and for what purpose, the responses may reveal:

WHY ⇒ “To support sample analysis for the Z-Line process”

“To demonstrate compliance with radiological control procedures”

WHAT ⇒ “Alpha, beta, gamma samples per sample analysis plan XYZ-99-1234”

WHOM ⇒ “For XYZ Division”

Identifying and documenting this information (such as in a systems engineering model) will capture this basis as justification for the task need. This exercise will help the systems engineer and the customer establish the boundaries of the problem or task. Weaknesses in the basis can also be examined to assure the initial task request is on solid ground and can stand up to scrutiny, if challenged. It will be shown later how this background will help refine the definition of the task and influence how the task proceeds.

A.2.2.2 2 Problem Definition and Problem Statement

Discussions are conducted with the customer and technical experts to better define and understand the problem that is being addressed and to assess the completeness of the input provided. Task requests generally present three possible scenarios or inputs: a problem is reported, a symptom is reported, or the customer requests something specific to be done (the “solution”). Since this input may or may not be complete or even address the real problem, it is necessary to gain a better understanding of why the request has been made to assure the real problem has been identified.

Ask the customer questions to assess the completeness of the input that was provided. The outcome of this questioning is an agreement with the customer on the problem to be solved, instead of symptoms to correct without solving the real problem. If the request is clearly a “solution” to an unstated problem, it is necessary to question the customer to identify the problem to be addressed.

Ask the following:

- Is this the problem or symptom of a problem?
- Should we be doing this task?
- Does it fix the real problem?
- Is this the best approach?
- Are the problem and task clearly defined?
- Who defined the problem and what’s their background?

This line of questioning will cause the customer and experts to rethink the task request and ensure that the problem the task is attempting to solve has been identified. In addition, the system engineer should also have the customer analyze conditions and identify and evaluate possible causes of the problem to determine a root cause. Identification of a root cause will help focus the problem statement.

To illustrate, consider the Q-Lab example. The initial task request to upgrade the Q-Lab facility is really a predetermined solution to an unstated problem. Therefore, what is the problem the customer is trying to address? Questioning may produce the following replies:

- “Existing equipment is old and unreliable.”
- “Results from Q-Lab do not meet QA accuracy requirements.”

The customer may then think that based on these “problems” the logical “solution” is his original request to upgrade Q-Lab. In reality, these “problems” are really symptoms of the real problem. By analyzing the conditions in Q-Lab the root cause for the problem surfaces:

- “We presently don’t have adequate capability to analyze the samples.”

The real problem in this case is more accurately stated as:

- “The current sample analytical capability will not satisfy Z-Line requirements specified in sample analysis plan XYZ-99-1234.”

The development work performed to establish the basis and rationale for the request (A.2.2.1), along with questioning the customer, provides the information needed to formulate an accurate problem statement. Obtain agreement with the customer that the problem has been accurately stated and document the problem statement. By correctly stating the problem, the potential for additional viable alternative solutions for consideration is introduced.

A.2.2.3 Mission Statement

By understanding the exact problem being addressed, a clear and complete mission statement for the requested task can be written. Essentially the mission statement captures the overall function the task must perform to satisfy the stated problem. In our Q-Lab example, knowing that the problem is that the current capability is inadequate, the mission statement can be stated as:

“Provide the analytical capability to perform sample analysis to satisfy the Z-Line process.”

This mission statement thus becomes a refinement of the task request. Notice that this is considerably different than the original request. As written, this mission statement opens up the possibility for other alternatives that could also satisfy the need, e.g., a new facility, perform the analysis elsewhere, share analysis with other labs, etc. It is also evident that the original task request to upgrade the Q-Lab is now one possible solution for consideration instead of the only solution.

A.2.2.4 Mission Goals and Objectives

Once the mission statement has been prepared, the overall goals and objectives for the task may be established. Often this effort will be a revision to the initial goals/objectives provided with the task request to better align them with the mission statement. The systems engineer and the customer should establish a mutually agreeable set of goals and objectives for the task.

Goals and objectives identify the desired conditions the customer would like to have achieved when the task is completed, and therefore, they provide a measure or "target" for performing the task. Unlike a requirement however, goals and objectives are those conditions that are desirable yet cannot be readily quantified or tested. For this reason, a goal or objective is a condition or end state that the task should strive to attain, yet it is not necessarily required to be achieved for the task to be successful. (Specific task requirements, developed later in the systems engineering process, will provide the measures for task success.)

Returning to the Q-Lab example, the customer may have originally stated a goal related to the completion of the requested upgrade to the Q-Lab. Instead, a more appropriate goal for the task may be:

"Maximize the capability to perform the sample analyses needed to maintain the Z-Line process operation."

The corresponding objectives are:

- "Increase the reliability of sample analysis methods"
- "Maximize efficiency of analysis operations"
- "Minimize sample turn-around time"

Again, goals and the objectives are related to the redefined task as clarified by the problem and mission statements.

A.2.3 Functions and Requirements

With a clear problem statement and mission statement prepared, and the task goals and objectives stated, the system engineer may now focus on developing the upper level functions and requirements that will shape the definition of the task. Refer to Appendix B, "Function and Performance Requirements Development" for guidance on the development and proper writing of functions and requirements. Again the customer's technical experts are instrumental in defining the requirements and the upper level functions that must be performed to achieve the mission and satisfy the problem. Once the task functions are known, the associated requirements for each function can be identified and linked to the functions. It is crucial that the basis and justification for each requirement be identified and documented.

Initially, the customer will have “drivers” or “originating requirements” for the task. Originating requirements are generally the requirements that surfaced when the basis and rationale for the task request were determined (see A.2.2.1). These requirements usually are very general in nature, but they provide the basis for the definition of the functions. The functions that are needed to satisfy these originating requirements are the upper level functions required for the task. These are the actions necessary to convert the initial conditions to the final desired state. The identified functions, in turn, may also prompt additional requirements that must be addressed, such as a performance requirement that is used to indicate the limits of the function.

The systems engineer works with the customer to assist in the proper identification and formulation of the functions and their definitions. If not intuitively clear, it is important to capture a precise definition of what the function means. The functions developed at the Mission Definition step only focus on the highest level, very broad functions that must be performed. It is not necessary to develop a detailed set of functions yet, and the systems engineer should keep the customer focused at a high level. Resist the temptation and natural inclination to drive down into increasing detail. Detailed functional analysis, performed later in the systems engineering process (refer to Appendix B), will decompose these functions into increasing levels of detail. Question the customer and the technical experts to ensure that all functions that must be performed to achieve the mission are identified.

The highest level functions can be considered an expansion of the mission statement, i.e., they provide additional clarity on what must be done to achieve the mission. In effect, the highest level functions are actually a decomposition of the mission statement. These functions better define the elements that must be considered by the task, and therefore provide an improved description of the scope of the effort. It is very important that these functions are not written based on a particular design solution. To illustrate, the Q-Lab example mission statement reads:

“Provide the analytical capability to perform sample analysis to satisfy the Z-Line process.”

This mission can be decomposed into several high-level functions that are necessary to achieve the mission:

"Receive Samples."

"Perform Sample Analysis."

"Operate Facility Infrastructure."

It can be seen from the example that the highest level functions include additional elements that must be part of the scope of the task to be successful. The function to perform the sample analysis is determined directly from the originating requirement for sample analysis for the Z-Line process. However, provisions must also be included to receive and handle the samples prior to analysis, and facility service systems (e.g., heating, ventilation, water, instrument air, etc.) must be available and operable as needed to support the analysis function. Note that the functions do not specifically favor or suggest any single potential solution.

Once the upper-level functions are identified, all originating requirements and any subsequently derived performance requirements are traced and linked to the appropriate function. This relationship defines and bounds the scope of the task and indicates the measures for success for each function. Performance requirements are derived from the customer's expectations for how well each function is to be performed. Each performance requirement must be stated in quantitative terms. For the Q-Lab example, the following examples of requirements may be identified and linked to the defined functions:

Function: "Receive Samples."

Originating requirement: "Receive alpha, beta, and gamma samples."

Function: "Perform Sample Analysis."

Performance requirement: "Analyze 50 samples per month."

Function: "Operate Facility Infrastructure."

Originating requirement: "Provide contamination control ventilation."

Performance requirement: "Hood ventilation air flow shall be a minimum of 125 linear feet per minute."

Again, it is not the intent to perform a detailed requirements analysis during Mission Definition. Instead the effort is limited to the originating requirements stated by the customer and any clarifying performance requirements, either given or derived, that help to define the expectations for each function.

A.2.4 Interfaces

The external interfaces for the task are documented to delineate the boundaries and specify the inlet and exit conditions for the task. The identification of external interfaces must include all pertinent interfaces. For the Q-Lab example, the external interfaces are the samples to be analyzed from the Z-Line process on the front end, and the sample analysis data on the back end. In addition, it is also necessary to include sample waste disposition as an exit interface.

A.2.5 Success Criteria

After the task has been thoroughly defined, and the functions and requirements have been identified, the criteria to be used to claim success are determined. Success criteria are the measures that the customer will use to judge whether the final state achieved by the task meets expectations and is acceptable. Question the customer and his or her experts to identify and specify the high level attributes and indicators that are important to the success of the overall task. As with requirements, these measures must be written in quantitative terms such that achievement can be determined. For the Q-Lab example success criteria might be:

- "Demonstrated ability to analyze samples within accuracy constraints specified in sample plan XYZ-99-1234."
- "Sample turnaround within the schedule requirements needed to support the Z-Line process."

A.2.6 Initial Risk Assessment

Potential risks associated with any aspect of the task should be identified and an initial assessment performed to determine if further evaluation is necessary as part of the task performance. Refer to Section 3.8, "Risk Analysis and Management," for guidance on evaluating risk. Any potential technical, cost, or schedule risks should be considered and subject to a risk screening. Any risk that could potentially have a significant negative impact to the completion of the task should be documented as part of the Mission Definition. A detailed risk analysis will be performed during the task to evaluate the severity of any identified risks and establish a plan for risk mitigation.

The screening of any identified potential risks relies on the expertise and judgement of the systems engineer, the customer's technical experts, and other subject matter experts. Since risks are inherent in any task performed, it is essential that serious consideration be given to identifying risks and properly screening the severity of the impacts due to the risks. Risks must never be downplayed.

Considering the Q-Lab example, a potential risk that may be possible is:

"New, untested analytical technology is necessary to analyze samples with the precision required by the Z-Line process."

This risk could have potentially serious negative impacts on the technical success of the task as well as on the cost and schedule. A detailed Risk Analysis will be necessary to manage the impacts associated with this risk.

A.2.7 Documentation

It has been noted repeatedly in this guide to document the information that has been generated. The importance of thorough, detailed documentation of the information obtained and developed during Mission Definition cannot be emphasized enough. Considerable effort has been spent to define and justify what has to be done to satisfy the customer's need. In addition, a significant amount of supporting information is developed and should be captured. This information forms an agreement with the customer on the exact scope to be addressed and establishes a baseline for the task.

Information may be documented by any suitable means. Simple text, tables, matrices, etc. may all be used as appropriate to capture and display task information. The use of specialized systems engineering software may be helpful to better document information on complex tasks. It is essential, however that the information is captured and presented in a manner the customer can use and readily understand. In all cases, have the customer review and concur with the Mission Definition prior to proceeding with the process.

A simple method for documenting the Q-Lab example used in this guide is shown in Attachment A.3.1. This example simply captures the information in a narrative style.

Attachment A.3.2 presents a sample of the use of CORE® to document the same information from the Q-Lab example used in this guide. The accompanying descriptive text that would be entered into the CORE® model for each element in Attachment A.3.2 would capture the detailed information that was generated. The software-defined relationships depicted in the diagram establish the links between the elements (e.g., functions, risk, requirements, etc.) entered into the model.

A.3.0 Attachments

A.3.1 Q-Lab Mission Definition Example

A.3.2 Q-Lab Mission Definition Using CORE®

Attachment A.3.1 - Q-Lab Mission Definition Example

Task Request:
Upgrade the Q-Lab Facility.

Basis:

Why:	To support sample analysis for the Z-Line process. To demonstrate compliance with radiological control procedures.
What:	Alpha, beta, gamma samples per sample analysis plan XYZ-99-1234.
Whom:	For XYZ Division.

Problem Statement:
The current sample analytical capability will not satisfy Z-Line requirements specified in sample analysis plan XYZ-99-1234.

Mission Statement:
Provide the analytical capability to perform sample analysis to satisfy the Z-Line process.

Goal:
Maximize the capability to perform the sample analysis needed to maintain the Z-Line process operation.

Objectives:
Increase the reliability of sample analysis methods.
Maximize the efficiency of analysis operations.

Minimize sample turnaround time.

Functions and requirements:

Function 1:	Receive samples
Requirement:	Receive alpha, beta, gamma samples
Function 2:	Perform sample analysis
Requirement:	Analyze 50 samples per month
Function 3:	Operate facility infrastructure
Requirement:	Hood ventilation air flow shall be a minimum of 125 linear feet per minute

Interfaces:

Input:	Alpha, beta, gamma samples
Output:	Sample analysis data Sample waste

Success Criteria:

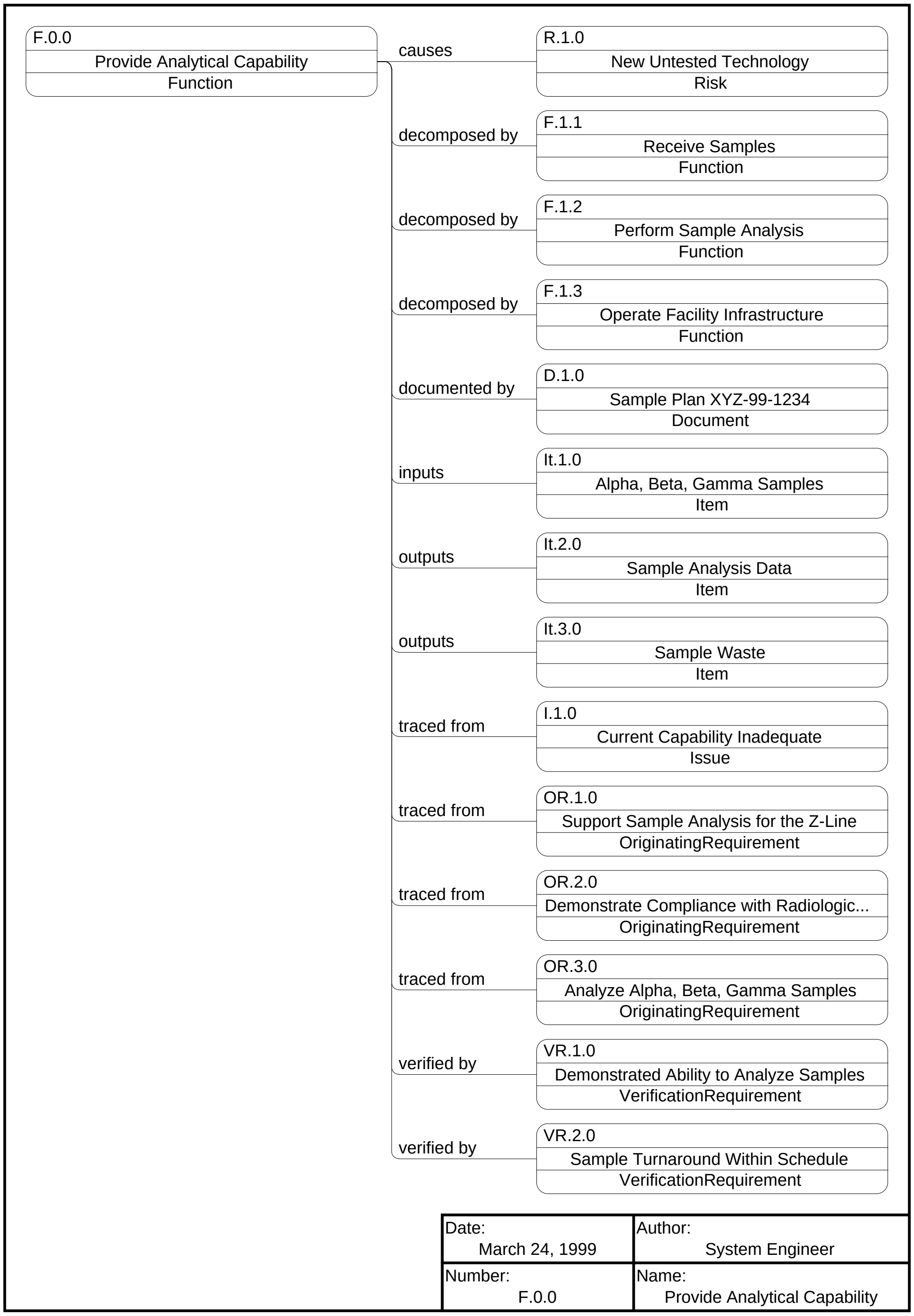
Demonstrated ability to analyze samples within accuracy constraints specified in sample plan XYZ-99-1234.

Sample turnaround within the schedule requirements needed to support the Z-Line process.

Risk:

New, untested analytical technology is necessary to analyze samples with the precision required by the Z-Line process.

Attachment A.3.2 - Q-Lab Mission Definition Using CORE®



Appendix B

FUNCTION AND PERFORMANCE REQUIREMENTS DEVELOPMENT

B.1.0 Introduction

This guide describes a process for the development of functions and performance requirements. The development of functions and performance requirements is at the heart of the systems engineering process. Functions describe what must be accomplished and performance requirements describe how well functions must be performed. Function and performance requirements development is not a standalone step but is instead one portion of the systems engineering (SE) process as a whole. This guide only addresses the function and performance requirements development portion of the process. Other guides provide assistance in completing the remaining system engineering process steps (e.g., Mission Definition, Functional Acceptance Criteria Development, Interface Control, Life Cycle Cost Analysis, Systems Engineering Management Plan Development).

B.1.1 What Are Functions?

A function is written most simply as a verb and noun combination (e.g., "filter particulates" or "measure temperature"). A function transforms inputs into desired outputs. For example, consider the function to "filter particulates." The function transforms an input containing particulates into two outputs, one with and one without particulates.

A function is a statement that provides a basis for a system to exist. It is a task, activity, or action that must be performed. What is the system there for? What does it do? A function describes what the system must do in order to meet the system's mission.

A more complete format for writing functions is to include the operating condition or accident / event when the function has to be performed. The suggested format for writing functions per the Writer's Guide for the Preparation of Facility Design Descriptions and System Design Descriptions (Reference D.3.2) is as follows: "(action verb and subject) during (operating condition or accident / event)." This additional information is necessary in order to clarify the function. Consider the "filter particulates" function again, does this function have to be performed under accident conditions or is it only required for normal operations?

If the function were written as “filter particulates during normal operations and all design basis accidents,” the purpose of the function would be more clearly communicated.

Every function has at least one performance requirement associated with it. A performance requirement quantitatively defines how well the function must be performed.

B.1.2 What Are Performance Requirements?

A requirement is something that the system must meet in order for it to successfully perform its mission. Requirements define the essential attributes of the system. There are three types of requirements; performance requirements, constraints, and interface requirements.

- Performance requirement - specifies how well a function must be performed
- Constraint - limits or constrains the design solution; these typically come from laws; regulations; DOE Orders; codes and standards; previous design decisions; operating / maintenance experience; etc.
- Interface requirement - requirement imposed on one system by another

Performance requirements are related directly to functions and are quantitative requirements of system performance. They specify how well, how fast, how much, how far, how frequent, etc. functions must be performed. Performance requirements are usually directly measurable (e.g., miles per hour, gallons per minute, feet, minutes). Consequently, every function must have a minimum of one performance requirement associated with it. Performance requirements control the overall system design by providing specific parameters that must be met by the design.

B.1.3 Why Are Functions and Performance Requirements Important?

Functions and performance requirements are developed as input to the design effort and their development is a key step in supporting project planning and definition. The process of function and performance requirements development focuses on describing the necessary and sufficient set of requirements that meet the mission need. By defining functions and performance requirements, the system purpose is clearly defined.

Functions and performance requirements are the key design input because they specify what and how well something is to be done. Clearly defined functions and performance requirements also enable planning of design activities and can assist in establishing system optimization limits. Engineers/scientists can always improve on something. However, when the functions and performance requirements are met, continued improvements are not necessary and should be stopped. When the design input, free of design solutions, is provided to system designers, it allows the designers to do their job with the most freedom, and to design the system that best meets the mission need. The functions and performance requirements provide the baseline to evaluate proposed designs. Consider Figure B1, Function/Requirement/System Relationships.

Figure B1 illustrates the relationships between the functions, requirements, and system architecture. Performance requirements, constraints (design requirements), and interface requirements are included in the figure. Everything is built upon the system functions.

Performance requirements are allocated to functions. This relationship identifies how well the functions must be performed. Functions are allocated to the system

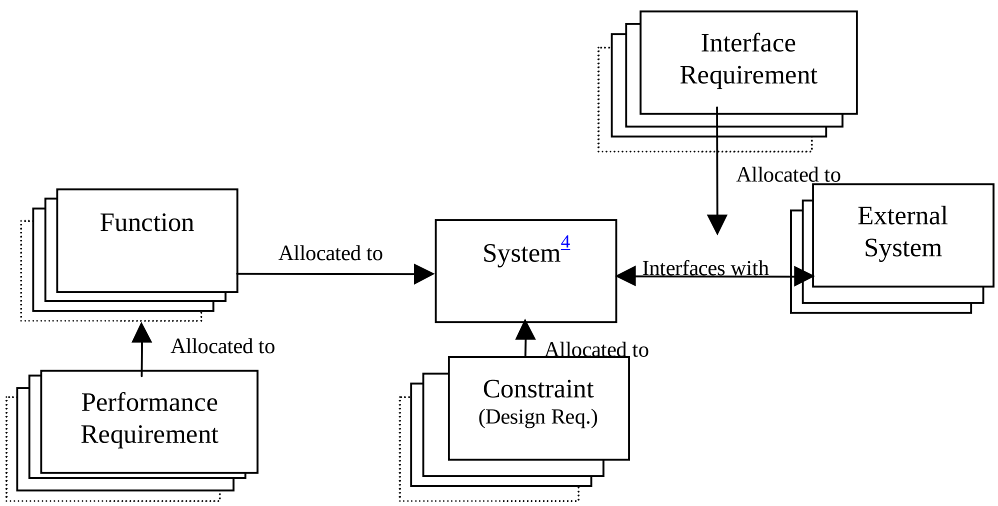


Figure B1. Function/Requirement/System Relationships

architecture. This relationship identifies what portion of the system architecture will perform the function. After the function to system allocation is made, the constraints can be completely identified and allocated to the system. That relationship indicates what constraints apply to what portions of the architecture. The figure also identifies the relationship between the system architecture, external system architectures, and the interface requirements. Interface requirements are identified and related to the interface between the systems.

The design process begins with identification of system functions and performance requirements. This indicates the importance of the function and performance requirements development process as the first step in preparing the design input on a task. Along with the functions and performance requirements, the known constraints and interface information (requirements and interfacing systems) are added as design input in order to more completely define the system. However, the system architecture must first be defined in order to completely specify the constraints and interfaces for a given layer of development (note that the physical architecture of the system is developed in layers and that each layer may have multiple levels within it.)

As an example, consider a function to "supply water" with performance requirements of a given pressure and flow rate. A constraint on the design may be known that requires water with given characteristics (e.g., domestic water versus service/process water). Based on the constraint requiring domestic water and the performance requirements of pressure and flow rate, the appropriate pipe codes for at least a portion of the system may be specified. However, for this example, two possible alternatives for supplying the water are 1) a holding tank system or 2) a connection to an existing header. Until the design selection has been made to utilize the holding tank system, for example, the selection of any pressure vessel codes for the holding tanks can't be made.

B.1.4 When is Function and Performance Requirements Development Performed?

The systems engineering process is iterative. The process begins with broad, task-related information lacking specifics and iterates toward increasingly detailed information. Each of the systems engineering process steps are performed at every layer of system development before proceeding to the next layer. The SE process steps are shown in Figure 2, the Systems Engineering Process Model.

Function and performance requirements development is performed during the Functions and Requirements Analysis and Allocation step (shaded in Figure 2). Function and performance requirements development is basically the process of converting the system mission analysis information into a well-defined, tangible set of actions (and associated requirements) the system must perform.

Figure 1 and the discussion in Section B.1.3 described in more detail how some of the elements and steps shown in Figure 2 are related.

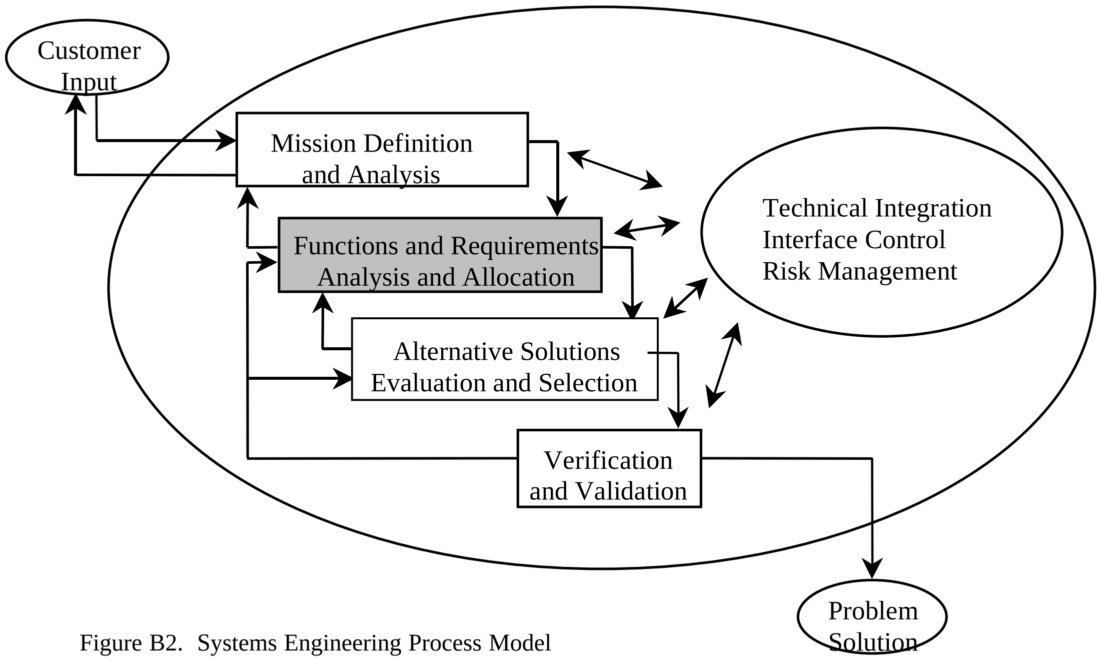


Figure B2. Systems Engineering Process Model

B.2.0 Function and Performance Requirements Development Process

Two general methods for functional development are presented in this section. The first method relies heavily on the identification of external interfaces. The external interfaces that cross the boundary of the system architecture are defined. The items crossing the system boundary are functionally traced through the system one at a time. The functions identified by tracing each item through the system, along with the functional interface information, are then combined to create a functional flow block diagram for the system.

The second method is driven by a hierarchical decomposition of upper-level functions. This hierarchically based method relies primarily on brainstorming by the functional development team as the means of identifying the lower level, more detailed functions that are required to perform the upper-level function being decomposed. Once the decomposition has been completed, the development team generates the functional flow block diagram and functional interface information.

Both of these methods are applied iteratively in conjunction with the other SE process steps. Each step in the SE process is completed at the most general layer of system development before moving down to layers with more and more detail.

A combination of these two methods is required at each layer in order to complete the functional development. Regardless of whether the functional development team begins with the first or second method presented in this guide, the other method needs to be applied in order to identify any holes or other problems. Table 1 and the discussion that follows provide a general comparison of the two methods.

The External Interface Method is more applicable as the starting point for functional analysis on an existing system or when several higher-level functions have been allocated to a system. This method is more easily applied when there are multiple functions at the upper level. This often occurs when working on existing systems because it's sometimes difficult to identify a single overriding function that is performed by an existing system. This method initially takes some of the focus off the upper-level functions and concentrates on the external interfaces. This method still requires that the lower-level functions decompose the upper-level functions of the system, but it becomes more of a test after the functions have been identified rather than the basis for the lower-level functions.

In contrast, the Functional Hierarchy Method places the majority of the focus on the upper-level functions. This method is more easily applied to a new system. The decomposition of the upper-level function generally results in no more than four or five functions being identified and a correspondingly simple flow diagram. The External Interface Method, on the other hand, tends to drive the functional analysis to a lower level of detail due to the tendency to get specific on interfaces. As a result, the External Interface Method better emphasizes system behavior and typically results in more complete functional flow block diagrams. The Functional Hierarchy Method makes it easier to stay at a higher level of detail when beginning a functional analysis for a new facility or system. Consequently, the more general functions that are developed with this method also allow for a simpler, cleaner allocation to system architecture. This can, correspondingly, result in making the development open to more alternatives and possibly provide a better solution.

External Interface Method	Functional Hierarchy Method
Lends itself to application on existing systems	Lends itself to application on new systems
Architecturally/physically based	Functionally based
Generally results in more functions and more detail for a given level	Generally results in fewer functions and less detail for a given level
More complicated allocation to physical components	Simpler allocation to physical components
Emphasizes system behavior and provides a complete picture on a single FFBD	Minimizes system behavior considerations due to multiple simple FFBDs
Doesn't guarantee all functions are identified, should be combined with Functional Hierarchy Method	Doesn't guarantee all functions are identified, should be combined with External Interface Method

Table 1. Function Development Process Comparison

With the Functional Hierarchy Method, each of the upper-level functions is decomposed and a separate functional flow block diagram is generated for each. This has the potential for not highlighting important functional interactions of an existing system, especially if the upper-level functions that have been identified are not very carefully considered. The External Interface Method generally results in functions from all of the upper-level functions being shown on one more complicated functional flow block diagram (similar to that of Figure A-8 in Attachment A). This facilitates a more thorough analysis of the functional interactions and can provide a more complete picture, although it can also lead to becoming bogged down in the details.

Often when the Functional Hierarchy Method is employed, the functional development team completes a functional decomposition for several levels of system development during one meeting or a series of meetings in a short time frame. This usually occurs without generating the accompanying functional flow block diagrams and N-squared diagrams or any of the other SE process steps. This is probably the biggest pitfall associated with employing this approach and should most certainly be avoided. As mentioned several times previously, EACH STEP OF THE SE PROCESS MUST BE COMPLETED AT EACH LAYER OF SYSTEM DEVELOPMENT.

Both methods have advantages and disadvantages and a quick application of “the other method” is required in order to double check results before moving on to the next step of the SE process. If the External Interface Method is applied to an existing system, apply the principles of the Functional Hierarchy Method to determine if the functional decomposition makes sense. Alternatively, when the Functional Hierarchy Method is applied, use the External Interface Method to determine if any holes exist in the functional decomposition.

B.2.1 Overview of External Interface Method

A summary of the External Interface Method presented in this guide can be found in Table 2, External Interface Method Summary. The table lists the process step and a brief description of the expected output from the step.

Process Step	Output
1-System Mission Analysis Review	Top level system functions and performance requirements
2-External Interface Identification	System external interface diagram
3-System Operational / Maintenance Concept Development	Narrative description of operational and maintenance concept, with system event list
4-Functional Sequence Development	System functional descriptions and simple functional flow block diagram with functional interfaces identified
5-Functional Sequence Integration	System functional descriptions and integrated, system functional flow block diagram with functional interfaces identified
6-Functional Hierarchy Generation	System functional hierarchy diagram
7-Performance Requirement Development	Performance requirement(s) for each function, with defendable basis

Table 2. External Interface Method Summary

The system mission analysis review involves simply gathering and becoming familiar with the output from the Mission Definition and Analysis step. The output from the Mission Definition and Analysis step is identified as: top level functions, top-level quantified performance requirements, initial risk assessment, external interfaces, and mission goals and objectives.

The external interface identification, as indicated above, should have been performed in the Mission Definition and Analysis step. Sometimes the external interfaces are identified during the mission analysis at a level that groups the items flowing across the interface at a level that is either too general or too detailed. In this case, this step involves adding some additional detail or aggregating the interface information. Otherwise, it is simply a review of the previously identified interfaces.

The system operational/maintenance concept development step is intended to initiate a discussion focused on the high-level vision associated with the system operation and maintenance. This step is highly conceptual and the descriptions produced at this point in the system development are likely to change, but these concepts lay the framework for the system behavior. This step forces the discussion and capturing of written concepts early so that all parties involved begin with a similar view.

The functional sequences are developed by identifying the functions that are performed by the system on items crossing the system boundary. This step involves identifying the functions that the system has to perform in order to transform the inputs to the system into the outputs from the system. This step results in a series of simple functional sequences.

The functional sequence integration step involves combining all of the simple functional sequences into one functional flow block diagram (FFBD). The system functional flow block diagram represents the system behavior, in its entirety, on one functional flow block diagram.

Following completion of the functional sequence integration, the system functional hierarchy diagram is completed. The functional hierarchy identifies the functional decomposition relationships.

The performance requirement development step results in at least one performance requirement being identified for each of the identified functions. The performance requirements must be quantified and have a defendable basis.

B.2.2 Overview of the Functional Hierarchy Method

A summary of the Functional Hierarchy Method presented in this guide can be found in Table 3, Functional Hierarchy Method Summary. The table lists the process step and a brief description of the expected output from the step.

Process Step	Output
1-System Mission Analysis Review	Top-level system functions and performance requirements
2-System Operational / Maintenance Concept Development	Narrative description of operational and maintenance concept, with system event list
3-Functional Decomposition	System functional descriptions and functional hierarchy diagram
4-Functional Flow Block Diagram Generation	System functional flow block diagram with functional interfaces identified
5-Performance Requirement Development	Performance requirement(s) for each function, with defendable basis

Table 3. Functional Hierarchy Method Summary

The system mission analysis review involves gathering and becoming familiar with the output from the Mission Definition and Analysis step. The Systems output from the Mission Definition and Analysis step is identified as: top level functions, top-level quantified performance requirements, initial risk assessment, external interfaces, and mission goals and objectives.

The system operational/maintenance concept development step is intended to initiate a discussion focused on the high-level vision associated with the system operation and maintenance. This step is highly conceptual and the descriptions produced at this point in the system development are likely to change, but these concepts lay the framework for the system behavior. This step forces the discussion and capturing of written concepts early so that all parties involved begin with a similar view.

The functional decomposition is developed by identifying those lower level functions that the system must perform in order to complete the upper level function(s). This step results in a functional hierarchy diagram.

The functional flow block diagram generation step involves identifying functional interfaces and capturing system behavior. The system functional flow block diagram and N^2 diagram or enhanced FFBD represents the system behavior.

The performance requirement development step results in at least one performance requirement being identified for each of the identified functions. The performance requirements must be quantified and have a defendable basis.

B.2.3 Performance Requirement Development

B.2.3.1 Function/Performance Requirement Relationship

As described in Section B.1.3, performance requirements are related directly to functions and are quantitative requirements of system performance. They specify how well, how fast, how much, how far, how frequent, etc. functions must be performed. Every function must have at least one performance requirement, although there are typically several, and the relationship between the functions and their respective performance requirements must be maintained. It should be very clear what performance requirements are associated with what functions. A simple numbering system may communicate this relationship.

An example numbering system is shown below. This sample numbering system makes use of a letter to differentiate the functions and performance requirements, "F" for function and "R" for performance requirement. The relationship between the performance requirement and its respective function is indicated by converting the "F" to an "R" and adding "*.x*."

F.1 "Description of function number 1"

- R.1.1 "Number 1 performance requirement statement"
- R.1.2 "Number 2 performance requirement statement"
- R.1.3 "Number 3 performance requirement statement"

F.2 "Description of function number 2"

- R.2.1 "Number 1 performance requirement statement"
- R.2.2 "Number 2 performance requirement statement"

Just as functions are decomposed into greater levels of detail, the accompanying performance requirements must be decomposed. Consider the example illustrated in Figure B3.

The upper part of Figure B3 illustrates the functional decomposition where upper-level function 1 is decomposed into three subfunctions, functions 1.1, 1.2, and

1.3. Also shown on Figure B3, are the accompanying performance requirements that are related to the same measure of performance. For example, upper-level performance requirement 1-1 may be a requirement limiting the total time allowed to perform upper-level function 1. The lower part of Figure B3 illustrates that the upper-level performance requirement 1-1 can also be decomposed such that the performance of the subfunctions to function 1-1 must be allocated to maintain the upper-level performance.

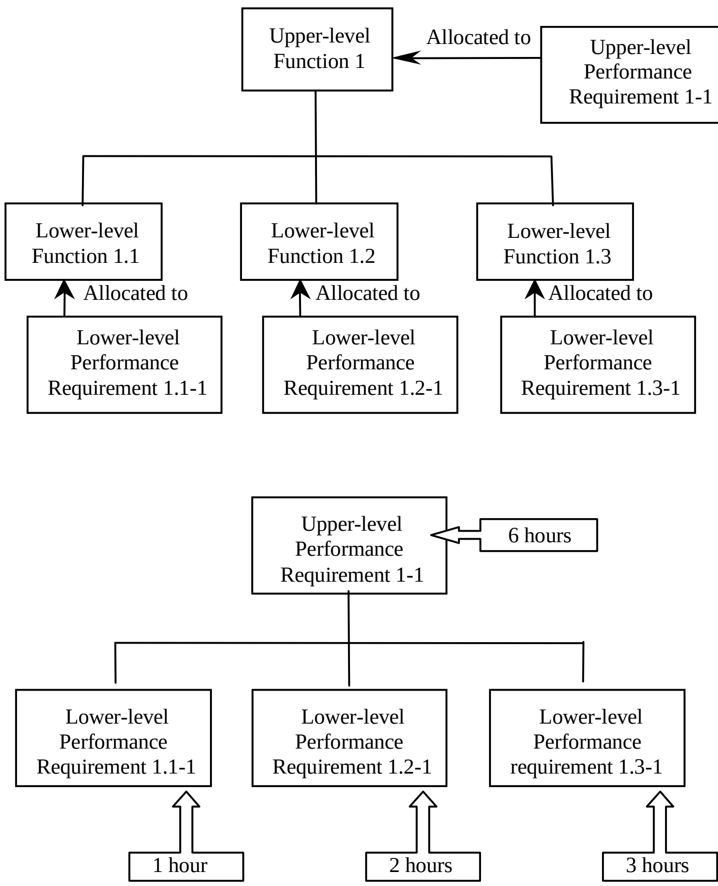


Figure B3. Example Performance Requirement Decomposition

For example, if upper-level performance requirement 1-1 is a time requirement such that function 1-1 must be performed in a maximum time of 6 hours, then the time for each of the subfunctions to be performed may be split into a maximum of 1, 2, and 3 hours for a total of 6 hours.

The above discussion on the decomposition of performance requirements does not mean to imply in any way that the subfunction performance requirements must be directly decomposed from an upper-level performance requirement. They must, however, still support the performance requirements of the upper-level function.

B.2.3.2 Developing Good (Performance) Requirements

This section provides guidelines for developing and writing good performance requirements. The guidelines are equally applicable to constraint and interface requirement development. Therefore, the more general "requirement" is referred to in the remainder of this section rather than the more specific requirement-type "performance requirement".

A list of key attributes of good requirements is provided below. A discussion of each of the attributes follows the list.

Key attribute list:

- Clear/concise, single-sentence format
- Necessary
- Attainable
- Verifiable
- Shall statements
- Defendable basis
- Implementation free
- Appropriate level
- Tolerances specified
- Positive format.

Clear concise, single-sentence format

Requirements should be written as a single sentence. This means that every requirement must be a standalone sentence with one requirement, stated clearly, simply, and concisely. One thought per requirement (per sentence) that ideally can't be misunderstood. Complex sentences with multiple clauses should be avoided. Each requirement should also be uniquely identified. Individual, uniquely identified requirement statements are necessary for traceability from higher level requirements, traceability to system functions or architecture, and for possible revisions.

Necessary

Every requirement has to be necessary. A requirement may be written clearly and concisely in a single, positive sentence as a "shall" statement, it may be free of design solutions, it may be quantified and specify tolerances, it may be written at the appropriate level, but if it's not necessary, it's still a bad requirement. This attribute of a requirement ties directly back to the basis for the requirement and illustrates the need to question and provide a defendable basis for every requirement. Asking "What is the worst thing that could happen if this requirement is not included?" is another good test for the necessity of a requirement. This question often results in identifying the requirement as being "nice to have" but not really a necessity and can often result in the requirement being converted to a goal.

Attainable

Every requirement must be attainable. As described above in the discussion for the necessity of each requirement, a requirement may possess all of the attributes that make it a good requirement, but if it's unattainable, it's still a bad requirement. A requirement may be unattainable for a number of reasons including technology, budget, schedule, or a higher-level requirement. If there are questions about the attainability of a requirement, feasibility studies may be required. Unattainable requirements may also be converted into goals.

Verifiable

Requirements should be verifiable. Every requirement must be written in a manner in which compliance can be demonstrated. Most often this becomes a problem when words like "maximize," "minimize," "to the maximum (minimum) extent possible," "user-friendly," "optimum," "sufficient," "adequate," "low," or "high" are used. Words specifying timing often create problems also. "Simultaneously," "quick," or "rapidly" mean different things to different people. Is simultaneous within 1 millisecond, 1 second, or longer? A helpful technique to employ when writing requirements is to ask the question, "How can this requirement be verified?" Requirements must be quantitative not qualitative.

A few more words or phrases that will cause problems when writing requirements are: "support," "and/or," "etc.," and "but not limited to." "Support" causes problems because it typically shows up in a requirement similar to this, "System *XYZ* shall support error recovery." The problem with this requirement is that it is open-ended and can't be verified. If there are certain functions that System *XYZ* must perform in order to support error recovery, then specifically list each function as a requirement with a defendable basis. Otherwise, this "requirement" may be converted into a goal that would feed into alternative studies as a decision criterion.

The problem with “and/or” isn’t really related to verification of the requirement but rather in realizing what it means when it is used. If “and/or” is used in a requirement statement where “*A* or *B*” is to be provided, then the requirement has been met if either A, B, or both A and B are provided. This isn’t a problem unless both A and B are required. Therefore, special caution is to be exercised if “and/or” is used.

When used in requirement statements, “etc.” and “but not limited to” result in requirements that can’t be verified, are surrounded by questions, and tend to leave things open to interpretation. They’re most often used in a list[1] and usually indicate that the author thinks there may be other items that haven’t been included. That may be so. However, by adding this element of the unknown in a requirement statement, the entire statement becomes unverifiable. Including “etc.” and “but not limited to” won’t cause additional requirements to be met should they happen to be identified at a later time, although including “etc.” and “but not limited to” may result in none of them being provided. As a result, these terms should be avoided. Just provide requirements for the items that are known and should additional items be identified later, revise the requirements.

Shall statements

Requirements must be written as “shall” statements. Requirements are not to use the word “should”. Requirements are things that must be met by the system. If a potential design solution doesn’t meet a requirement, it is no longer considered a design solution without rework (either to a requirement or to the potential solution).

“Should” is typically used when writing goals. A goal is to be clearly differentiated from a requirement. A goal is something that is to be strived for given other requirements. Goals are direct input for decision criteria in alternative analyses and trade studies. Goals provide a basis for evaluating potential design solutions. Additionally, “will” refers to statements of fact and must not be used when writing requirements.

[1] Note that use of lists is not recommended. As noted in this Section, each requirement should be uniquely identified. When items are included in a list, there is usually not a unique identifier for each separate item. One exception where this may be acceptable, is the case where each item has the same basis and will be verified by the same test. This rarely happens and, as such, the use of lists is discouraged.

Defendable basis

Every requirement must have a defendable basis. The basis includes the supporting rationale for the requirement. The basis references any data, trade studies, or other sources for the requirement. Any assumptions made that resulted in the requirement and the associated logic should also be provided in the basis. The basis is typically included in an appendix with design input documents. This is an acceptable format to enable easier reading after review and approval of the document. However, it is recommended that, at least for any early drafts, the basis be included with the requirement statement. This facilitates the review by eliminating any flipping back and forth between an appendix and the body of the document, emphasizes the importance of the basis, and helps to ensure the basis is indeed reviewed.

Implementation free

Requirements must state what the system does rather than how the system must do it. A common pitfall when writing requirements is to specify a design solution rather than the requirement behind it[2]. To avoid this problem ask, “Why is this requirement needed?” If that question doesn’t take you back a level, then the requirement is probably stating the need rather than the implementation. Asking this question commonly results in a number of separate requirements replacing the original “design solution” requirement statement. This question also helps to identify the basis for the requirement once the design implementation has been removed. Other than the obvious problem with specifying a design solution, that of potentially eliminating a better solution, there is a potentially more dangerous problem. The second more dangerous problem is that of assuming that specifying a design solution covers your actual needs. This may result in a product delivered as specified that does not deliver what is required. Another problem associated with specifying a design solution in a requirement statement comes about when the verification is performed. If there is a verification that the “design solution” requirement has been met, the only thing that has been verified is that the system has a design, not that the design works. This effectively results in eliminating any value added from verification activities.

---

[2] Requirements are often generated in order to fill a perceived hole in a requirements document. This common practice tends to lead to the specification of design solutions rather than requirements and great caution should, therefore, be exercised if this practice is undertaken.

- Example: Consider the following requirement statement written into an aircraft specification—"The aircraft shall have three engines." This is clearly a requirement specifying a design solution. When the question "Why do you need three engines?" is asked, the real requirement that the aircraft shall be able to operate with an engine failure would become apparent. It is also easily seen that requiring three engines rather than requiring that the aircraft operate with an engine failure could result in the real requirement not being met.

- Another common example of stating implementation is demonstrated with the following requirement—"The Container Transport Subsystem shall control position to within ±0.5 inches in three dimensions." This example requirement indirectly constrains the system design by specifying a subsystem. One last example requirement stating implementation rather than the real need is given by the requirement, "A database shall be provided." When the question "Why is this requirement needed?" is answered, the following 'real' requirements are given; "The capability for traceability between items shall be provided," "The capability to add attributes to items shall be provided," "The ability to sort items shall be provided."

Appropriate level

An additional caution related to including implementation in the requirements is specifying requirements at an appropriate level. Recall that the SE process is iterative, it runs through each of the basic SE process steps at a given layer. After a layer is completed, the next lower layer of development begins. When specifying requirements, it is important to keep in mind what stage, or layer, of development the system is in. If the requirements are being developed at the system layer, requirements should not be included for individual components. Specifying lower-level requirements at the upper levels of system development tend to overly constrain the design and are an indirect way of specifying implementation. As a general rule, if the requirement does not apply completely to the scope, or piece, of the system that you are currently working on, it should instead be included at a lower layer. In other words, requirements should be specified at a layer where they affect all the parts below that layer. This is a rule that intends to place the focus on the bigger picture before moving into the details.

- As an example, think about developing requirements for a facility in a FDD. Requirements that are specific to an individual component or system should not be included in the FDD. Instead, the requirement should be specified in the appropriate SDD (in the component section, if it's applicable to an individual component as opposed to the entire system).[3]

Tolerances specified

Requirement tolerances should be specified. Requirements written without tolerances can quickly lead to increased costs, both from a product delivered without the required tolerances as well as those with unnecessarily tight tolerances. It's pretty obvious what kind of problems you can get into when close tolerances are required and aren't provided. But the opposite can be true as well. For example, consider the requirement to "...provide a lifting capacity of 1,000 lbs." Imagine, for this example, that other requirements restrict this lifting function to a forklift and that there are no readily available commercial (and theoretically cheaper) forklifts available with a lifting capacity of less than 2,500 lbs. The requirement specifying a lifting capacity of 1,000 lbs may result in a special-purpose design for performing the lifting function because it is unclear whether a 2,500 lb. capacity forklift is acceptable. If the requirement were written instead as "...provide a minimum lifting capacity of 1,000 lbs." then it is clear that the 2,500 lb. capacity forklift would be acceptable.

Positive format

Requirements should be written in a positive format. Requirements written negatively are, at a minimum, difficult to read and understand, and can sometimes be impossible to verify.

- Example: "The function shall not be completed in more than 10 seconds." As written, this example is difficult to read and understand. This requirement should be written as "The function shall be completed in less than 10 seconds." Consider another example, "The system shall not allow failures due to operator input." This is an example of a typical "shall not" requirement that is impossible to completely verify. This type of requirement statement should be avoided.

---

[3] This often causes problems for both the customer and system development teams because of a fear that a requirement will be lost. However, if the requirements are captured at an inappropriately high layer, they end up being repeated at the lower layer, or they get changed, or they disappear altogether due to development at the upper layer. If a requirement gets repeated in a lower-layer requirements document, there are duplicate requirements that must be managed. If the requirement is changed or disappears, this forces a revision to the upper-layer requirements document. The use of a holding bin for requirements that come up but really don't belong at the level where work is being performed is suggested as an effective tool for avoiding this situation.

B.3.0 Suggested Reading List

Writing Good Requirements, Hooks, I. (1994), Proceedings of the Third International Symposium of the NCOSE.

Systems Engineering and Analysis, Blanchard, B.S., and Fabrycky, W.J.(1990), Prentice Hall, Inc.

System Requirements Analysis, Grady, J.O. (1993), New York: McGraw Hill, Inc.

Discovering System Requirements, A. Terry Bahill and Frank F. Dean, located at http://www.sie.arizona.edu/sysengr/requirements/index.html.

Characteristics of Good Requirements, Pradip Kar and Michelle Bailey, Proceeding of the Sixth Annual International Symposium of the International Council on Systems Engineering, July 7-11, Boston, MA, located at http://www.incose.org/workgrps/rwg/goodreqs.html.

Guide for Managing and Writing Requirements, Hooks, I., 1994.

Appendix C

ALTERNATIVE STUDIES and VALUE ENGINEERING

C.1.0 Introduction

C.1.1 Purpose

The purpose of this guide is to describe the steps, tools, and techniques involved in the Alternative Study (aka Trade Study) Process (including value engineering) as integrated into the application of the Systems Engineering Process for DOE activities. This guide is to be used for selecting the optimum, most cost-effective alternatives that meet an activity's functions and requirements. While the major application of alternative studies (in particular value engineering) is in design and construction projects, these activity alternatives can be in other areas such as operations, maintenance, administrative processes, etc.

C.1.2 Types of Studies and Distinctions

There are numerous methods available for evaluation of an activity and selection of the best method to accomplish the activity. Such tools include cost-time profile evaluations and process improvement analyses for ongoing operations and processes, benchmarking for new ventures, carbon copy facility design for new constructions to eliminate variability and capitalize on lessons learned, use of engineering judgment, etc. This guidance document covers the alternative study method, including a specific type of alternative study – namely, value engineering - and the recommended methodology for application.

C.1.2.1 What is an Alternative Study?

An alternative study is a tool used to select from two or more options available to meet a specific function. Alternative studies encompass analysis of functions and are directed at optimizing performance, reliability, quality, safety and life-cycle cost of a product or activity. Alternative studies include the following steps:

- Identification of the function(s) to be met and the defined project requirements
- Identification of alternatives that perform the function(s)
- Determination of viability of the alternatives to satisfy requirements
- Establishment and weighting of criteria against which to evaluate alternatives

- Evaluation of alternatives against the selected criteria
- Selection of a preferred alternative.

An optional step, employed when no alternative is clearly preferred based on the results of the alternative study, is a sensitivity analysis.

Attachment 1 provides a flow chart of the alternative study process.

Alternative studies can be conducted at one of three levels:

- A simple, informal alternative evaluation. This process is suggested where no alternative poses a significant risk to the success of the activity. One alternative is clearly superior to all other choices and/or there are no discernible criteria for selecting among alternatives. An informal memo may be used to document the selection of the alternative and basis for the selection.
- An informal alternative study. This study follows the same process as a formal alternative study, but less rigor is applied to the conduct of the study and documentation of the results. This process is suggested where the risk to the activity, based upon the selection of any of the alternatives, is moderate, the activity is not complex, and discernible criteria can be identified. The study may be documented in a memo, incorporated in other documentation for the activity, or presented in a formal report.
- A formal alternative study. This process follows a structured approach for comparison of alternatives. The process uses formal analysis and is based on a set of weighted decision criteria. This process should always be applied to line-item projects and other complex activities or where the risk to the activity, based on the alternative selection, is relatively high. A formal alternative study is documented in a formal report.

C.1.2.2 What is a Value Engineering Study?

A value engineering study is a specific type of formal alternative study that follows a prescribed methodology or job plan. In addition to optimizing performance, reliability, quality, safety, and life-cycle cost, a value engineering study is specifically intended to identify solutions that improve upon these features relative to an established baseline. Value engineering studies are led by an individual trained in value engineering and conducted interactively by a team, selected jointly by the manager of the activity being studied and a VE-trained individual, who are independent of the work performed on the program, project or activity. While an alternative study can follow the defined methodology for a value engineering study, all value engineering studies must complete certain steps to be considered as value engineering studies.

The steps followed in conducting value engineering studies are:

Information Gathering – The step in which the value engineering team collects information relative to the activity. Most often, cognizant activity personnel initiate the information gathering process with a technical presentation several weeks in advance of the interactive study session. This step includes generation of a Functional Analysis System Technique (FAST) diagram by the value engineering team and culminates in identification of those functions which, by design, may have a Cost/Worth ratio higher than that necessary to meet requirements. Attachment 2 provides details on completion of a FAST diagram and Attachment 3 discusses Cost/Worth ratios.

Creativity or Speculation – The process of generating alternative potential methods for accomplishing a given function.

Analysis and Judgment – The process of evaluating identified alternatives. This step includes development and weighting of criteria against which alternatives can be measured, and determination of the relative merit of an alternative against those criteria for the purpose of selecting the optimum alternative(s).

Development – The process of defining details associated with the selected alternative(s). These details include a description of the alternative and a comparative analysis between the selected alternative and the baseline, including a cost estimate for the selected and baseline alternatives.

Recommendation/Presentation – Identification to decision makers of recommendations resulting from the value engineering study.

As with other alternative studies, a sensitivity analysis is often included in the analysis phase to assure proper selection of a preferred alternative.

C.1.3 When to Perform an Alternative Study

C.1.3.1 Scope

All decisions made during the conduct of an activity include an alternative evaluation in some form. However, not all evaluations of alternatives require a documented alternative study. The depth and formality of the alternative study are dependent upon the complexity of the decision being made (see section C.1.2.1

above). A documented alternative study should be conducted when criteria can be established that discriminate among potential alternatives, especially when it is unclear if or how all alternatives meet the identified functions, or when there is a significant difference among the alternatives in terms of risk to the activity. A formal alternative study is selected when the activity is complex or risks are considered high.

C.1.3.2 Timing

There is no specific timing recommended for conducting an alternative study that covers all cases. Since all decisions involve an alternative evaluation, alternative studies are conducted as needed throughout the activity.

C.1.4 When to Perform a Value Engineering Study

C.1.4.1 Scope

A value engineering study is intended to apply a level of independence to an activity and the selection of steps to complete this activity. Value engineering is conducted when numerous functions are assigned to the activity and their integration and interrelationships are complex, when significant financial resources are required to support the activity, when criteria selection and weighting are subject to interpretation, or when the evaluation of alternatives could be implemented and interpreted in several ways. In general this applies to all line-item projects.

C.1.4.2 Timing

Unlike other alternative studies, value engineering studies begin with a baseline approach or design. In addition, since a value engineering study can result in recommending some significant changes in project direction, it is recommended that the study be conducted before significant effort has been devoted to design detail. For these reasons, the optimum timing for a value engineering study is between the completion of the conceptual design and the initiation of the detailed design. Attachment 4 illustrates the potential for realizing benefits from a value engineering study at various phases of the project cycle.

C.2.0 Methodology and Tools

There are a number of different methods available to facilitate conduct of a value engineering or alternative study. Several of these are discussed in the following sections.

C.2.1 Study Initiation and Information Gathering

Alternative studies are generally initiated during the normal course of work for new constructions, modifications, and projects any time a decision is required. Often a conceptual design report identifies a number of critical areas where the need for documented alternative studies is envisioned. For other activities, initiation of an alternative study is based upon a perceived need on the part of users to evaluate various ways to meet their requirements.

Because alternative study participation is intended to rely on individuals involved in and knowledgeable of the activity under study, the need to provide an orientation meeting to initiate the study is limited. Generally, only individuals brought in as study facilitators or subject matter experts require background information in advance of the study.

Because a Value Engineering Study Team is expected to be independent of the activity being studied, the planning needs associated with value engineering studies differ somewhat from those of other alternative studies. Prior to initiating the value engineering interactive study, the Study Team must be provided with information regarding the activity. This information is to include the technical information regarding the design and/or operation, as well as a cost estimate of the design, maintenance, and operations. For efficiency, personnel expert in the activity being studied (e.g., Project/Design Teams, Maintenance/Operations personnel, etc.) should provide this information to the Study Team approximately two weeks in advance of the study.

C.2.1.1 Functions and Function Analysis

The first step in an alternative study is function identification and analysis. In the majority of alternative studies, this step involves a list of one or more functions required to meet user needs. Sometimes these functions are decomposed to greater levels of detail, but generally are limited in scope to a defined study topic (e.g., system design alternatives, component selections, etc.). While there is no difference in the function analysis process between informal and formal alternative studies, informal alternative studies generally include fewer systems and components and consequently fewer functions due to the lower level of risk. In value engineering studies this step culminates in a Function Analysis System Technique (FAST) diagram (see Attachment 2). While function definition is a critical part of the systems engineering process, FAST diagramming differs in the following ways: FAST diagram preparation is done independent of the design effort; FAST diagrams are done by a team of individuals who did not participate in the design decisions to date; FAST diagrams follow a “HOW-WHY” logic;

FAST diagrams are constructed to a level of detail commensurate with the needs of the study, not to the level of detail required to complete design work.

Unlike function generation and decomposition in design, where functions and requirements are defined first and design solutions that meet these functions and requirements selected next, FAST diagrams are based on the functions of the structures, systems, and components already identified in the design.

C.2.1.2 Cost/Worth Evaluations

The cost/worth evaluation is a comparison by the Study Team of the lowest cost available to meet a given function (the "worth" of the function) against the actual identified cost for the structure, system, and/or components selected to meet this function (the "cost" of the function). Note that cost/worth ratios have little meaning if there is no proposed design or if a cost estimate has not been prepared for the proposed design. Thus, cost/worth ratios are most commonly associated with value engineering studies, that rely on the existence of a baseline approach than with other alternative studies.

Some caution is required in developing cost/worth ratios. Many items, especially structures, systems, and complex components, are designed to accomplish multiple functions. Cost estimates, however, are usually available no lower than the component level. Thus the cost of a specific function is only a part of the cost of the component. The Team must judge what portion of the component cost is attributable to the specific function. This value is often, at best, a judgment call on the part of the Team. Similarly, the worth of a function is the Team's best guess of the least expensive method available to meet the function.

Often it is sufficient for the purposes of a value engineering study to identify that the cost/worth ratio is ">>1," ">1," "=1," or, in some cases, "<1." Functions with higher cost/worth ratios are the prime targets for value improvement.

C.2.2 Generation of Alternatives: Speculation

Generation of alternatives is usually done through Team brainstorming. In many alternative studies a list of alternatives to be considered is identified outside the interactive Team setting, generally as a result of initial design considerations or by user (facility) or DOE prescription. As with function analysis, there is no difference in the process for generating alternatives between informal and formal alternative studies, although informal alternative studies generally have fewer functions, thus a lesser scope, resulting in fewer applications of the process (although not particularly in identification of fewer alternatives for each function identified). In value engineering, alternative generation is always done as a part of the interactive Team setting.

In Team brainstorming, high-cost/low-worth functions are first identified. The Study Team spontaneously produces various ideas on how to perform the identified function. Creative, divergent thinking is essential in this step. No ideas are to be critiqued at this stage and all ideas are recorded. Critical comments at this point tend to inhibit the flow of ideas. Furthermore, even frivolous suggestions can result in successful recommendations. For example, to meet a certain function a Team member might say "Let Superman squeeze it". While this may seem absurd, it could lead to a successful suggestion of using pressure, or a pressurized system, to perform a function when temperature control was previously used.

C.2.3 Evaluating Alternatives: Analysis and Judgment

Often the speculation phase results in one or both of the following: a number of infeasible alternatives, and a number of mutually exclusive alternatives. In the analysis phase, the Study Team must evaluate alternatives for both feasibility and selection of the best alternative from among several. Alternatives are evaluated for feasibility by ensuring first that they can perform the required functions and, second, that they meet the stated requirements. If the alternative fails either of these tests, it is eliminated or revised to perform the functions and meet the requirements. The best alternative is selected by establishing criteria against which to measure the various alternatives, choosing the relative importance of these criteria (i.e., weighting the criteria), and measuring the alternatives against the weighted criteria. These steps are discussed below.

C.2.3.1 Selecting Criteria

- Short List of Criteria

  Generally, once a list of alternatives has been developed, there are an extensive number of choices for meeting the functions identified. At this point it may be prudent to narrow this list to a manageable number. To do this a "Short List" of decision criteria may be employed. The short list identifies criteria that often represent "GO/NO GO" factors, as identified by activity requirements such as technological feasibility or the capability to produce a given quantity per unit time. In this case, alternatives that can not be designed to meet the requirements of the project are eliminated. Caution must be exercised in eliminating alternatives using GO/NO GO criteria so as not to eliminate alternatives that could be made viable. For example, if production rate requirements are 1,000 tons per year, based upon written requirements, any alternative producing 999 tons per year or less is eliminated. Users must be sure that requirements do not have a margin that includes the capabilities of the given alternative or that can not be legitimately modified to allow inclusion of the alternative.

► Decision Criteria

Criteria selection ultimately determines the alternative choice. Identification of criteria can be a simple task for a Study Team or it can be quite complex with numerous decisions included in the selection. Care must be taken to ensure that the criteria selected allow for discrimination among alternatives, i.e., if the color of all alternatives is the same or the user is indifferent to the color selection, then color is not a criteria. Although no requirements exist relating the quantity or type of criteria to the depth of the alternative study, criteria are typically selected that are commensurate with the level of risk associated with the activity being studied. Thus, informal alternative studies, which are expected to have a lower associated risk, usually have fewer, less complex criteria than formal alternative studies. Alternative performance must be capable of being measured or estimated for each of the decision criteria selected. This may be more involved for formal alternative studies, but must be commensurate with the level of effort applied to the study and the phase of development of the alternatives. For example, if alternatives are currently in the preconceptual phase of development and a decision criteria is selected as “maximizing performance y”, the effort required to estimate how the alternatives score on the criteria shouldn’t require a 3-year research and development program.

When an alternative study is being performed on a project, the project’s mission analysis should be the primary source for generating decision criteria. These criteria should be based on the project goals, objectives, requirements, and DOE and other stakeholder values.

Decision criteria should:

— Differentiate between alternatives

— Relate to project goals, objectives, and values of DOE and other stakeholders

— Be reasonably measurable or estimable

— Be independent of each other

— Be well understood by all decision makers.

There are several methods available to facilitate criteria selection. The first method is team brainstorming. In this approach all Team members spontaneously voice their opinion of criteria and all opinions are recorded. This method has the advantage of allowing all Team members to identify their ideas in an impromptu manner, minimizing prejudgment. The disadvantage of this method is that quieter members may never express their opinions.

A second method is round robin. In this approach, Team members are individually asked for their input of criteria. Again, all inputs are recorded. This method has the advantage of soliciting input from all Team members. However, it provides members an opportunity to prejudge what they are thinking and tends to thwart creativity.

A third method is reverse direction criteria development. In this approach, Team members consider some alternatives available, identify differences between these alternatives and develop criteria that reflect these differences. This technique is most useful when the viable alternatives, inclusive of their "pros" and "cons," are well known.

Because the criteria selection process relies heavily on human judgment, criteria development is done manually (i.e., without the aid of computer applications). However, a predefined set of criteria may be provided from external sources such as end-users, stakeholders and decision-makers, for incorporation into the final set. Input from the decision makers is essential to the development of the criteria set.

Once a full set of criteria has been established, these criteria can be modeled into a hierarchical parent-child relationship. Attachment 6 provides an example of this modeling process. Although application of this modeling is not restricted, it is more commonly useful with complex, high-risk decisions. Thus, this is generally applied to formal, but not informal, alternative studies. Hierarchical modeling of criteria facilitates both establishment of criteria weights and evaluation of alternatives against the criteria (see Sections 2.3.2 and 2.3.3). Duplicate criteria, or criteria that do not discern among the alternatives, should be eliminated.

C.2.3.2 Criteria Weighting

Although weighting of identified criteria is not required for all alternative selection processes (see Section 2.3.3), in complex decisions it is difficult to justify a single solution without consideration of the relative importance of the criteria established for making the decision.

Criteria weighting can be accomplished in several different ways:

- Direct decision and input of constant values for criteria weights
- Weight Ratios and Analytic Hierarchy Process
- Partial Weight Ratios
- Weight computation through ordering importance
- Weight computation based on "swing weights"
- Weight computation through tradeoffs of alternatives.

Each of these methods is described below:

Direct Decision and Input of Constant Values for Criteria Weights

The simplest way to weight criteria is through direct input of criteria weights. These weighted values predominantly come directly from decision makers, are established through expert judgment, or a combination of these. In this method, once the criteria have been selected, decision makers/experts decide how important each criterion is as a percentage of unity. Each criterion is given a relative score of between 0 and 1 (or 100%), depending upon its importance in selecting an alternative from among several. All criteria receive weights, with the total of these weights being 1 (or 100%). This method does have noted disadvantages; it can be difficult to reach Team consensus using this method. Furthermore, the method can introduce additional bias into the judgments over those introduced by other weighting methods.

Weight Ratios and Analytic Hierarchy Process

Another method for weighting criteria is the weight ratio (WR) methodology. WR methodology uses pair-wise ranking and "relative value" methodology to weight criteria. Each criterion is compared to each of the other criterion one set at a time. In comparing the criteria sets, Team members decide which of the two criteria is a more important factor in selecting an alternative and by how much.

The WR process can be completed either manually or via the use of various computer software tools available. In the simplified manual method, Team members collectively agree on which criterion in a given pair is more important and on the value for this relative importance. The scale for "how much" is numeric and is determined by the Team, although scales of one to five and one to ten are well recognized. In the latter case, one represents equal importance of the criterion and ten represents an order of magnitude difference between the two criteria.

Once established, this relative value score is summed for each criterion and is then either normalized to a scale of 0 to 10 or converted to a percentage, with the total of all scores being 100%. Attachment 7 provides a template and example of manually generated criteria ranking.

Advantages of simplified manual pair-wise comparisons are that, for a small number of criteria, it can be completed quickly during the interactive session. Disadvantages of this method are that one of the identified criteria should always go to a score of "0," thereby eliminating its influence on the decision. Consistency checks must be done separately (i.e., if A>B and B>C then either A>C or A>>C should be true). With larger numbers of criteria, total consistency is difficult to achieve and very difficult to check.

The Analytic Hierarchy Process (AHP) uses a specialized application of the WR methodology. In AHP, again individual criteria are compared one set at a time. In this comparison, Team members either collectively agree on which criterion is the more important and by how much, or individual members "vote" on these comparisons. In AHP, a criteria scoring range of one to nine is used. When individual voting is used, a single final score is established by using the geometric mean of the individual scores.

Equation Figure

The geometric mean is defined by:

s_i = individual score of a pair-wise comparison;

GM = geometric mean

For this application, the geometric mean is simply the n^{th} root of the product of n individual scores. Its value may be demonstrated for cases where one or more scores are widely dispersed from the rest. For example, in the set [1 2 3 9], the average, or arithmetic mean, is 3.75, while the geometric mean is 2.711. In this case, the arithmetic mean is greater than 75% of the individual elements. By using a geometric mean, the impact of widely varying perceptions on the relative importance of criteria is minimized. AHP then proceeds by using matrix mathematics and the eigenvector solution to establish criteria weights.

An advantage to AHP is that in AHP all criteria receive a score - i.e., if criterion A is 4 times more important than criterion B, then criterion B is ¼ as important as criterion A. Both numbers are used in the calculations. Thus, no criterion weight becomes zero, as with the simplified WR method.

As with the simplified application of WR, criteria weighting using the AHP methodology can be performed manually. Attachment 8 provides detailed instructions for establishing the weighting matrix and the use of the eigenvector solution to determine criteria weights. It is recommended, however, that if manual application is desired, the simplified WR methodology be employed.

Several software tools are available for automated implementation of WR methodology. Among them, the software tools Expert Choice (ECPro®) and Logical Decisions®, both of which apply AHP, are comparable and are relatively easy to use. An advantage of software-support use of AHP is an internal consistency check of the value comparisons.

Partial Weight Ratio

The partial weight ratio method utilizes pairwise comparisons as in the AHP process except that only enough pairwise comparisons are completed to ensure that each criterion has been included at least once. Because this method relies on an abbreviated set of criterion comparisons, no manual method is presented. This process is, however, supported through the Logical Decisions® software tool. An advantage of this method is that it is somewhat quicker to implement than AHP and can be utilized when evaluation Team members are uncomfortable comparing certain criteria. However, a disadvantage is that without all pairwise comparisons, a consistency check of inputs is not possible.

Weight Computation Through Ordering Importance

In the weight computation through ordering importance method, Team members define an alternative with the least preferred level of acceptability against all criteria. Team members then select the one criterion they would choose to improve, given this choice. This criterion becomes the most important criterion. The process continues until all criteria have been ranked. This method offers an advantage when comparison of criteria on a one-to-one basis is difficult. A disadvantage of this method is that criteria ranking is established on a mathematical interpretation of “preferred” criterion. Thus all weights are established on a binomial selection process rather than a relative value process.

Since success of this method is based upon a mathematical relationship established between “preferred” and “next preferred,” etc. criteria, it is recommended that this method, like weight computation, be utilized through available software. Logical Decisions supports this process.

Weight Computation Based on Swing Weights

Weight computation based on “swing weights” is a combination of ordering preference and direct decision and input. In this method, as with ordering preference, Team members define an alternative with the least preferred level of acceptability against all criteria, then select the one criterion that they would choose to improve. This criterion is then given a “swing weight” of 100. Team members then similarly select the next criterion and determine the relative importance of “swinging” it over its range compared with swinging the first criterion over its range, as a percentage of the first criterion’s 100 point swing weight. The process continues until all criteria have been ordered. The advantages to this method are similar to those for ordering preference, except that criteria ranking is adjusted to reflect the evaluators’ judgments on relative criteria importance. A disadvantage is that the idea of relative importance of swinging criteria through their range is rather abstract and could be difficult for individuals to implement.

This method is implemented by adjusting the absolute weights to sum to one. This can be done manually or via supporting software. For large matrices it is suggested that, as with ordering preference, a software tool be used. Logical Decisions supports this process.

Weight Computation through Tradeoffs of Alternatives

In the weight computation through tradeoffs of alternatives method, two alternatives of equal preference are identified. This method is based upon the idea that equally preferred alternatives should have equal utilities. In this method, Team members identify pairs of equally preferred alternatives that differ on exactly two distinct criteria, C1 and C2. The tradeoff begins with each of the two alternatives receiving the best value for either C1 or C2, and the minimum for the other criterion. Alternative 1 receives the best value for C1 and the worst value for C2 and alternative 2 receives the best value for C2 and worst value for C1. (The alternatives have equal values for the remaining criteria.) In performing the tradeoff, team members start by identifying which of the two alternatives is most preferred. Is alternative 1, with the best value for C1/worst value for C2, preferred or alternative 2 with the best value for C2/worst value for C1? Assuming alternative 1 is preferred, the team members would identify the value change in C1 required to bring alternative 2 to an equally preferred value to alternative 1. The inputs are mathematically manipulated through the relationship Weight(C1) x Value change(C1) = Weight(C2) x Value change(C2) to establish relative weights for the criteria. The disadvantage to this method is that it requires a mathematical input for the value, and the change in value of an alternative against the two criteria. This information may be difficult to develop. Certain software tools, however, allow this to be performed graphically. Again, the software tool Logical Decisions supports this process.

Table 2.3.2 summarizes the various weighting methodologies described here, their limitations and strengths, and suggests potential applications appropriate for each.

Table 1. Criteria Weighting Methodologies Summary

Methodology	Limitations	Strengths	Recommended Uses
Direct Decision & Input	More prone to introduction of individuals' biases	Simple - No evaluation team effort required to select and weight criteria. Incorporates high level decisions not otherwise apparent to evaluators	When Decision Makers have expertise to determine relative importance of criteria
Weight Ratio – Simplified	Eliminates low importance criteria Consistency check of data is difficult, especially with large quantity of criteria Less conducive to hierarchy structure of criteria	Allows fast completion of criteria weighting in interactive session	When few criteria exist, criteria are independent of each other and criteria hierarchy structuring is not needed When least important criterion will not influence alternative selection
Weight Ratio – AHP	May need availability of software for efficiency of implementation in some applications Requires hardware and data inputting when results are required during interactive session	Accommodates numerous criteria, some of which are derived from others (Hierarchy Structure) Conducive to inputting some direct decision values and adjusting others accordingly	When sensitivity evaluations are desired When activity is complex When consequences of decision result in high risk to activity
Partial Weight Ratio	Can not perform check on consistency of individual's data See also Weight Ratio – AHP	Eliminates need for criteria comparisons that are difficult	When evaluators have difficulty with comparison of several criteria
Ordering Importance	Does not use relative values of criteria to determine weights Requires alternative with lowest score against all criteria	Direct comparison of criteria is not required Faster than AHP for interactive sessions	When a one-to-one comparison of criteria is not feasible
Swing Weights	Requires more time than Ordering Importance Abstract concept	Direct comparison of criteria is not required Conducive to expert/ decision maker inputs	When a one-to-one comparison of criteria is not feasible When more representative weighting is desired
Tradeoffs of Alternatives	Requires thorough knowledge of two available alternatives which are equally preferred Requires numerical alternative values	Compares criteria against an example	When alternatives are equally preferred but for different reasons When more representative weighting is desired

C.2.3.3 Alternative Selection

As with criteria weighting, selection of a preferred alternative can be done either through a manual or a software-assisted process. There are a number of recognized methods for selection of a preferred alternative. Six of these methods are described below.

Discussion of Pros and Cons

Almost invariably in an evaluation of multiple alternatives each alternative being considered has distinct advantages (pros) and disadvantages (cons) as compared to the other alternative(s). In this method these pros and cons become the criteria against which the alternatives are evaluated. For simple, minimal risk, non-complex, alternative evaluations in which the pros and cons are distinct among the alternatives, an acceptable method for selecting the preferred alternative is a general presentation and discussion of these pros and cons. Although weighting of these pros and cons is not required, the discussion should include a justification as to why the pros of the selected alternative are more important and the cons of less consequence than those of the other alternatives.

As an example, assume that the objective is to construct a new secondary road. Given alternatives of asphalt and concrete, the pros and cons listed are:

Table 2. Evaluation of Multiple Alternatives

	Pros	Cons
Asphalt	Lower capital cost Lower maintenance cost	Less durable
Concrete	More durable Higher capital cost	

In this case, since the lower maintenance cost of the asphalt would offset the durability of the concrete, an ensuing discussion would justify selecting asphalt based upon estimated usage and overall life cycle cost (capital plus maintenance costs).

Since this method presumes simplicity of the activity being studied, as well as the alternatives under consideration, the method is typically only used in informal alternative studies.

Nonweighted Criteria Method

This method for selection of a preferred alternative from among several choices involves the development and use of criteria. These criteria, however, are not weighted and is only slightly different from the pros and cons method described above.

In this method, a list of criteria is established, usually developed as a result of the evaluators' knowledge of the advantages and disadvantages of various alternatives. These criteria are then listed on one side (either the vertical or horizontal) of a matrix. Identified alternatives are listed on the other side. Each alternative is then evaluated against each criterion and assigned a comparative ranking. This ranking can be numerical or otherwise representative of the differences (e.g., +, -, 0). The alternative with the most positive score(s) becomes the preferred alternative.

As an example, consider again construction of a secondary road. If the previous alternative selection set of asphalt and concrete were expanded to include a dirt road, cobblestone, and brick, and criteria of "capital cost," "maintenance cost," "durability", "ride quality," and "aesthetics" were developed, a matrix could be generated as follows:

Table 3. Method for Selection of Preferred Alternative

	Asphalt	Concrete	Dirt	Cobblestone	Brick
Capital Cost	0	-	+	-	-
Maintenance Cost	+	-	0	0	0
Durability	0	+	-	0	0
Ride Quality	+	+	-	-	-
Aesthetics	0	0	0	+	+

From this matrix all alternatives except asphalt appear to average a neutral or lower score against the selected criteria. Asphalt averages a moderate + score. Thus, asphalt would be the preferred alternative.

An intuitively obvious disadvantage to this method is the lack of the relative importance of the criteria. Thus the usefulness of this method is greater when all criteria are relatively equally important or when the selection of an alternative is more a matter of simply making a choice and the resultant decision is essentially risk free.

Dominance Method

The dominance method compares all criteria of one alternative to another, as follows:

If the scores for all the criteria for one alternative are higher than these scores for another alternative, then the former alternative is said to dominate the latter. Because all criteria scores for one alternative are higher than those for the other alternative(s), this method does not require that the criteria be weighted. The alternative determined to be dominant becomes the preferred selection. This method is most useful when there are an exceptionally large number of alternatives and relatively few criteria, in that one alternative usually does not score higher than another alternative on all criteria, especially once the "less feasible" alternatives are eliminated. Although this method may be useful in reducing the number of alternatives, it usually will not yield a single preferred alternative.

Sequential Elimination Method

The sequential elimination method considers one criterion at a time to examine alternatives for elimination.

1. The alternative with the highest value for the most important criteria is chosen. If a number of alternatives perform equally well, they all remain viable.

2. The remaining alternatives are sequentially evaluated for each criterion, in order of descending importance of the criteria, until only one alternative remains. This alternative becomes the preferred selection.

Although this method is viable, its application is extremely limited in that it does not consider all criteria concurrently, and in fact, generally neglects those criteria with lower importance.

Minmax Method

The minimax method is initiated by having Team members identify, for each alternative, that alternative's lowest score against any of the criteria. The Team then determines which of the low scores is the highest. The alternative with the highest of the low scores becomes the preferred alternative.

As with other methods, this method may not definitively select an alternative. In addition, this method has the disadvantage of only considering each alternative's weakest criterion, independent of the relative importance of the criterion against the other criteria. Since, predominantly, the lowest criteria value for each alternative comes from different criteria, the comparisons are based on dissimilar standards.

Scoring Method

Use of a scoring method is the preferred technique for evaluating alternatives and selecting a preferred alternative. In the scoring method the merit of each alternative is determined by summing the contributions to that alternative from each identified criterion. In this method, weighted criteria must be used if the criteria have varying degrees of importance. In the scoring method of alternative selection, defined and weighted criteria are used to select the optimum from among a set of alternatives that meet the defined function. A simplified example of this process is provided in Attachment 9.

Aside from the simplified application provided in Attachment 9, two of the most common scoring methods for alternative selection are Multi Attribute Utility Theory (MAUT) and the Analytic Hierarchy Process (AHP). Either of these processes can be done manually, although the mathematical manipulation of data can become cumbersome. Generally both of these tools are applied with the assistance of software tools. The tool ECPro supports AHP, while the tool Logical Decisions supports both MAUT and AHP.

The foundation of the MAUT is the use of utility functions. These utility functions are intended to allow comparisons on a one-to-one, "apples to apples" basis for diverse decision criteria. Every decision criterion in the alternative study has a utility function created for it. The utility functions serve to transform the diverse criteria to one common, dimensionless scale or "utility." Once the utility functions are created, alternative raw scores can be converted to a utility score and then they may be compared with each other and an alternative score totaled for all the criteria.

The utility function converts an alternative's raw score against a given decision criterion to a normalized utility score which reflects the decision maker values. For example, assume that one of the decision criteria in an alternative study is to minimize cycle time and another is to minimize the amount of liquid waste generated. For this example alternative study, three alternatives have met all the requirements and are considered feasible. The following table shows the raw scores for the alternatives against the two decision criteria.

Table 4. Raw Scores for Alternatives by Decision Criteria

	Alternative A	Alternative B	Alternative C
Cycle Time (hours)	3	6.5	4
Liquid Waste (gallons)	22	5	15

Alternative Raw Scores

The figures below illustrate two possible utility functions for the two decision criteria. The range of utility values is typically from 0 to 1, but can be any range as long as it is consistent for each decision criterion. It can also be seen from the figures that the best raw score for each criterion is usually assigned the value of 1 and the worst raw score a value of 0. In this case, a 3-hour cycle time would receive a utility score of 1, and an alternative that generates 22 gallons of liquid waste would receive a score of 0.

The utility function for the cycle time is represented by a straight line indicating that the value system of the decision makers is directly correlated to the cycle time. That is to say, an increment of 1 hour is valued the same at the lower end of the cycle time range as it is at the higher end of the range (going from 4 to 3 hours cycle time has the same value in utility as going from 6.5 to 5.5 hours). The example utility function for the liquid waste criterion, on the other hand, represents a nonlinear relationship between "utility value" and gallons of waste produced.

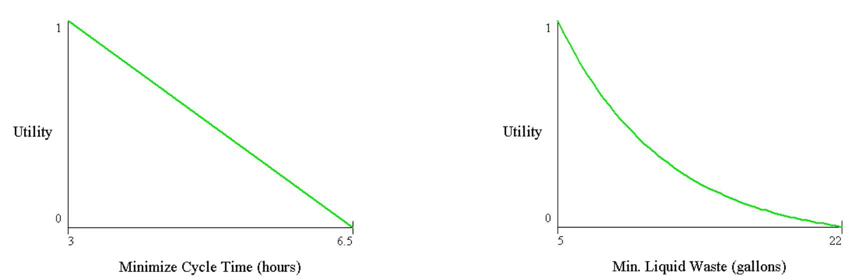


Figure C1. Utility Functions for 2 Decision Criteria

With this nonlinear utility function, an increment of one gallon of waste produced has a different utility at each end of the liquid waste produced range. This utility function indicates that moving from 5 to 6 gallons causes a significantly larger drop in utility score than moving from 21 to 22 gallons. This, in effect, says that the decision makers value an alternative that produces a small amount of waste much higher than one that produces waste at the larger end of the range.

Once the utility functions are generated and the raw scores are converted to utility scores for each of the alternatives, the utility scores can be converted to a weighted utility score (by multiplying the utility score by the weight of the decision criteria) and totaled for each alternative. See Attachment 10 for an example of alternative evaluation using the MAUT method.

The use of utility functions is typically employed when more information is known about the alternatives, resulting in firmer estimates of the alternative performance. However, the MAUT method can be employed when the alternative scoring is more subjective. When this is done the utility function is generated in the form of an analytic expression. This provision is especially helpful when detailed estimates of alternative performance are available for a portion of the criteria but several criteria remain more subjective. In this case, the alternative study should maximize the use of the well-developed information by utilizing the MAUT method with analytic expressions for some of the criteria.

Assume in the example given above that the cycle time criterion was less developed and actual estimates for the alternatives did not exist. In this example the higher level driver for the criterion was to minimize the total time it takes to completely stabilize a given type of material. Instead of knowledge about the cycle time for the process, assume that the decision makers had a more subjective feel for the time required to stabilize the material under each alternative (see table below).

Table 5. Alternative Scores

	Alternative A	Alternative B	Alternative C
Material Stabilized by	End of FY 2001	End of FY 2008	End of FY 2003
Liquid Waste (gallons)	22	5	15

Alternative Raw Scores (More Subjective)

An example utility function utilizing analytic expressions is provided in the table below for the minimize stabilization time criterion:

Table 6. More Subjective Alternative Scores

Utility Score	Expression for Alternative Performance
1	Material will be stabilized by the end of fiscal year 2001
0.5	Material will be stabilized by the end of fiscal year 2003
0	Material will be stabilized by the end of fiscal year 2008

Subjective Utility Function Example

With this utility function, Alternative A would receive a utility score of 1 and Alternatives B and C would receive utility scores of 0 and 0.5, respectively. It should be noted that, as in this example, when using this type of utility function a nonlinear value system may be applied. This function could have been created to represent a linear relationship between the utility score and time to complete stabilization.

When applying the MAUT method to the more subjective criteria, it is recommended that the descriptions of alternative performance be as detailed as possible and that a minimum of four or five utility scores be described. This will allow for a more consistent scoring to be applied to each of the alternatives. This is especially important when a large number of alternatives are being considered and when a large number of decision makers are evaluating the alternatives.

These examples presented a small number of possible utility functions. For more examples of utility functions see Attachment 11. As previously described for assigning weights to decision criteria, there are numerous methods for generating utility functions. Attachment 11 also provides a description of some of the methods for generating utility functions supported by the Logical Decisions software.

AHP uses "ratio values" rather than pure utility functions in selecting a preferred alternative. AHP does not require explicit levels for measures, although any of the measures can be defined based upon quantitative inputs. In this methodology, a preferred alternative is selected using pair-wise comparisons of the alternatives based on their relative performance against the lowest-level criteria in the hierarchy structure (see Attachment 6). The evaluation, or weighting of alternatives, is similar to the process defined for weighting criteria (see Attachment 8) - i.e., against criterion A, which alternative, 1 or 2, is better, and by how much – 1x, 2x, … 9x? This results in alternative preference weights for each of the lowest-level criteria. These alternative preference weights are then multiplied by their respective criteria weights and summed to produce overall alternative preference scores, with the highest score being the preferred alternative.

The major disadvantage of AHP, as perceived by some, is the fact that the process relies upon expert judgements of the decision-makers, both in prioritizing criteria and selecting a preferred alternative, using subjective pair-wise comparisons. Proponents of AHP, on the other hand, view this subjectivity aspect of the process as a definite positive in that it utilizes the knowledge base of the decision-maker.

Table 7 summarizes alternative selection methodologies, and their uses and limitations.

C.2.3.4 Sensitivity Analysis

In general, preference for one alternative is considered clear if the score for the preferred alternative exceeds the score for any other alternative by 10% or greater. In some instances this does not occur. In these cases a sensitivity analysis is recommended.

The purpose of a sensitivity analysis is to validate the alternative evaluation and ranking of alternatives that result from the decision process by demonstrating that small changes do not change the alternative ranking. These small changes could occur for the alternative scores against the decision criterion, decision criterion weights, or requirements.

The sensitivity analysis should evaluate the impacts of adjusting alternative scores up and down by approximately 10%. The Decision Team should insert raw score changes of ± 10% for each of the alternatives against the decision criteria. If these small changes don't change the overall results, then the analysis is insensitive to the alternative scores.

After verifying insensitivity to the alternative scores, the decision criteria weights should be checked for sensitivity. Once again, the Decision Team should make changes of ± 10% for each of the decision criteria weights while maintaining the 100% sum of the weight factors. If these changes don't result in a change in the alternative rankings, then the decision analysis is considered insensitive.

Making minor changes in the requirements is another possible check for sensitivities in the analysis. This could allow additional alternatives to qualify for the analysis by passing any go/no-go gates. This exercise is suggested when there are alternatives close to any requirement cutoffs.

Table 7. Alternative Selection Methodologies Summary

Methodology	Limitations	Strengths	Recommended Uses
Discussion of Pros and Cons	Relative importance of Pros and Cons not readily apparent. Limited to small number of criteria.	Simple to implement	Lesser risk applications, few alternatives, easily discernible pros/cons
Non-Weighted Criteria Method	Relative importance of criteria not readily apparent	Simple to implement. More conducive to higher number of alternatives and criteria than Pros and Cons method	Lesser risk applications
Dominance Method	Requires that one alternative be superior to another against all criteria. Does not typically result in a selected alternative	Quickly eliminates alternatives which could not be selected	To eliminate alternatives from a long list before performing a more formal alternative study.
Sequential Elimination Method	Ignores less important criteria. Does not consider alternative performance against all criteria.	Can be implemented quickly	When the highest one or two criteria dominate the decision as to which alternative will be selected
Minimax Method	Does not typically result is a selected alternative. All but one criterion are ignore: This criterion is different for each alternative	Can be implemented quickly	When all criteria are relatively equal in weight and alternatives are closely grouped in performance against the criteria.
Scoring Method – Simplified	Relative comparisons of alternatives against each criterion are fairly subjective. Limited to a small number of criteria.	Can be done in interactive session without computer hardware and software. Relative criteria value is considered	When the relative relationship among alternatives with respect to the criteria is clear and assigned values represent the differences
Scoring Method – MAUT	Requires development of and agreement with utility functions. Requires more well developed info on alternative performance.	Relative comparison of alternatives is the least subjective of any of the methods. Results in best understanding of decision maker values.	For complex, high risk decisions requiring easily interpreted and defensible results with well developed alternatives.
Scoring Method – AHP	Typically requires computer hardware and software for efficiency of effort. Relative comparison of alternatives will be a linear relationship	Does not require utility function. Relative criteria value is considered.	For complex, high risk decisions with less developed alternatives under consideration.

If any of these steps in the sensitivity analysis result in changes to the ranking, the Decision Team should reevaluate the criteria, alternative scores, or requirements that resulted in the sensitivity. This step is meant to ensure that the values and weights given to the element that caused the sensitivity are appropriate and that the team understands the impact that the element has on the decision. Following completion of the sensitivity analysis, confidence in the alternative rankings should be established.

It should be noted that the majority of the software available for decision making allows for sensitivity analyses to be performed very simply. Both the Logical Decisions® and ExpertChoice® software generate excellent graphs to analyze the decision sensitivity and both also allow for dynamic sensitivity analysis.

C.2.3.5 Special Case Criteria Development

Often the selection of an alternative is based upon criteria that are not straightforward or conclusive. In these cases, it may be required to evaluate the alternatives against these criteria using a "subordinate" supporting analysis or model. Some examples of this include:

1. Life-Cycle Cost Analysis

   Life-Cycle Cost analysis is used to evaluate the relative costs of the alternatives. Life-Cycle Cost analyses provide the following types of information:

   — Cost information for system effectiveness

   — Cost of development, manufacturing, test, operations, support, training, and disposal

   — Design-to-cost goals, any projected change in the estimate of these costs, and known uncertainties in these costs

   — Impacts on the life-cycle cost of proposed changes.

2. End-Product and Cost-Effectiveness Analysis

   End-product and cost-effectiveness analysis is conducted on system processes – generally life cycle processes – including such features as test, distribution, operations, support, training, and disposal. These analyses support:

   — Inclusion of life cycle quality factors into the end-product(s) designs

— The definition of functional and performance requirements for life-cycle processes.

3. Environmental Analysis

   Environmental analysis is used to identify and ensure compliance of the alternative(s) with all federal, state, municipal, and international statutes and hazardous materials lists that apply to the activity. These analyses include environmental impact studies to determine the impact of an alternative during the life cycle; on the infrastructure; on land and ocean, atmosphere, water sources, and human, plant, and animal life. Subcriteria in these analyses include such things as avoiding use of materials or the generation of by-products that present know hazards to the environment, and enabling integration and synchronization with activities that support NEPA documentation (e.g., Environmental Impact Statements and Environmental Assessments).

4. Risk Analysis

   Risk analysis is performed to identify the impact of undesirable consequences, based upon the probability of occurrence and consequences of an occurrence. The results of the risk analysis are prioritized and used as input to the alternative study.

5. Economic Analysis

   Economic analysis is conducted to eliminate as many cost biases as possible. An economic analysis involves evaluating all known costs of an alternative, from preconceptual activities through decommissioning.

6. Modeling and Optimization

   Modeling is used to facilitate an alternative study by describing a system via a simplified representation of the real world that abstracts the features of the situation relative to the problem being analyzed. There are four types of models in use: physical, analog, schematic, and mathematical.

C.2.4 Development of Results

Following selection of a preferred alternative, the Study Team develops details regarding the selected alternative to support the results. This supporting detail includes:

(1) Identification of the specific alternative or alternative features considered, inclusive of a thorough, but concise, description of these alternatives emphasizing those features that differ among them

(2) Advantages and disadvantages of the preferred alternative over the other alternatives

(3) A life-cycle cost comparison among the various viable alternatives, generally recommended as a relative life-cycle cost comparison in lieu of a complete life-cycle cost analysis.

Not all studies, nor all recommendations within a given study, require the same level of detail in developing the recommendations. The appropriate level of detail is that which is necessary and sufficient to justify the recommendations. Studies conducted at earlier stages of a project generally have less concrete quantitative data available than those conducted following conceptual or detailed design. Often costs are in "order of magnitude" terms and operations and maintenance costs are based on industry standard values for a given facility type or size. Studies conducted during construction and operation should contain a significant level of detail regarding cost differentials, including actual operations and maintenance cost comparisons, to justify changing an activity at that stage.

C.2.5 Presentation and Reporting of Results

With the exception of simple mental selection alternative studies, the results of all value engineering and alternative studies should be formally reported.

C2.5.1 Written Report of Study Results

Following completion of a value engineering or alternative study, the Study Team documents the results. For informal alternative studies this is often done as a part of another document. Formal alternative studies are typically documented in stand-alone reports. This documentation includes:

(1) Description of process/methods used

(2) Function analysis and/or functions against which alternatives were identified

(3) Identification of the various alternatives proposed, inclusive of a concise description or descriptive title of these alternatives

(4) Identification of the criteria and criteria weighting used to select the preferred alternative, including a description of the meaning of the criteria

(5) Identification of preferred alternative, including alternative evaluation against the criteria

(6) Development documentation of the preferred alternative (see section 2.4)

(7) Dates and time of the study

(8) Study participants and their past involvement with the activity.

A suggested report outline, intended to assure inclusion of this information, is provided in Attachment 12.

C.2.5.2 Oral Presentation of Study Results

Following completion of a value engineering or alternative study, one or more Team members may prepare a formal presentation for management/decision makers identifying the recommendations for changes to the activity under study. For value engineering studies, this presentation should clearly and concisely identify the "before" activity and the "after" or recommended activity, the advantages and disadvantages of implementing the proposed change, and a relative cost comparison between the proposed activity and the baseline activity. Alternative study presentations should clearly identify the various alternatives considered as well as the advantages and disadvantages of the alternative selected and, if available, a cost comparison among the alternatives evaluated. It is recommended that the Team member championing a given proposal present the proposal.

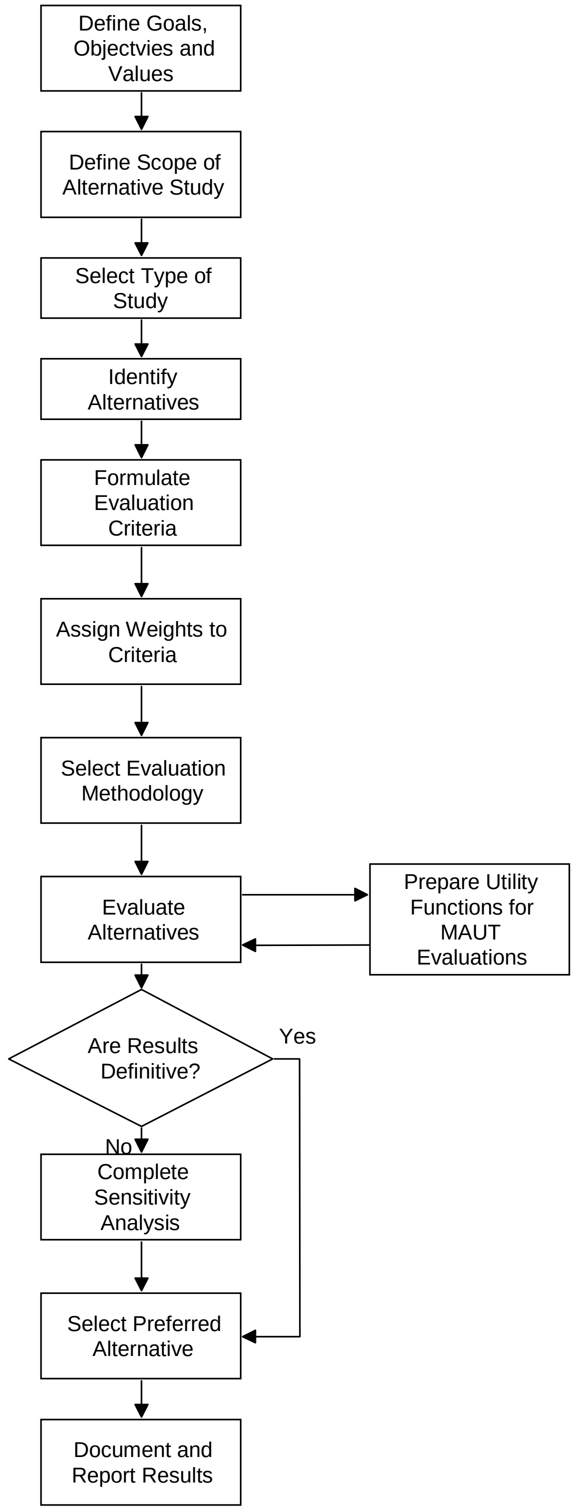
C.3.0 Study Closure

Closure of alternative studies and value engineering studies differ in a number of ways. Since participation by project personnel is expected in conducting alternative studies, an alternative study is considered complete when the study report is signed by a responsible manager within the area being studied (e.g., the Modification Manager or Project Manager for modifications/projects, the Facility Manager for activities affecting operations, etc.) or, in the case of alternative studies documented within another document, when the governing document is signed. Responsibility for implementation of any recommendations included in the study resides with this signature authority individual.

Because a value engineering study is conducted independently from the personnel responsible for the activity, these studies are not approved by cognizant activity personnel. Instead, these studies are only approved by the authoring personnel. In this instance, a formal transmittal letter is sent to the cognizant activity personnel requesting that they disposition the recommendations. Although a part of the value engineering report is documentation of potential cost savings, the cognizant activity personnel are expected to identify any actual cost, especially those which result in a budget change.

ATTACHMENT 1

ALTERNATIVE STUDY PROCESS FLOW CHART

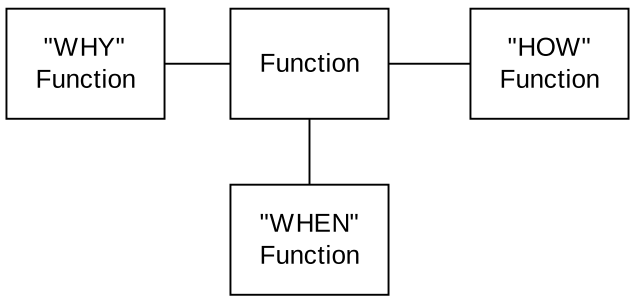


ATTACHMENT 2

FAST DIAGRAMMING

FAST diagramming is the creation of a logic structure of functions associated with a system, using a “HOW and WHY” relationship. A function immediately to the right of any other function on a logic path describes HOW the function is achieved. A function to the left of any other function on the path describes WHY the function is performed. A function directly below another function on the path identifies that the function on the path is accomplished WHEN the lower function is accomplished. The figure below illustrates this relationship.

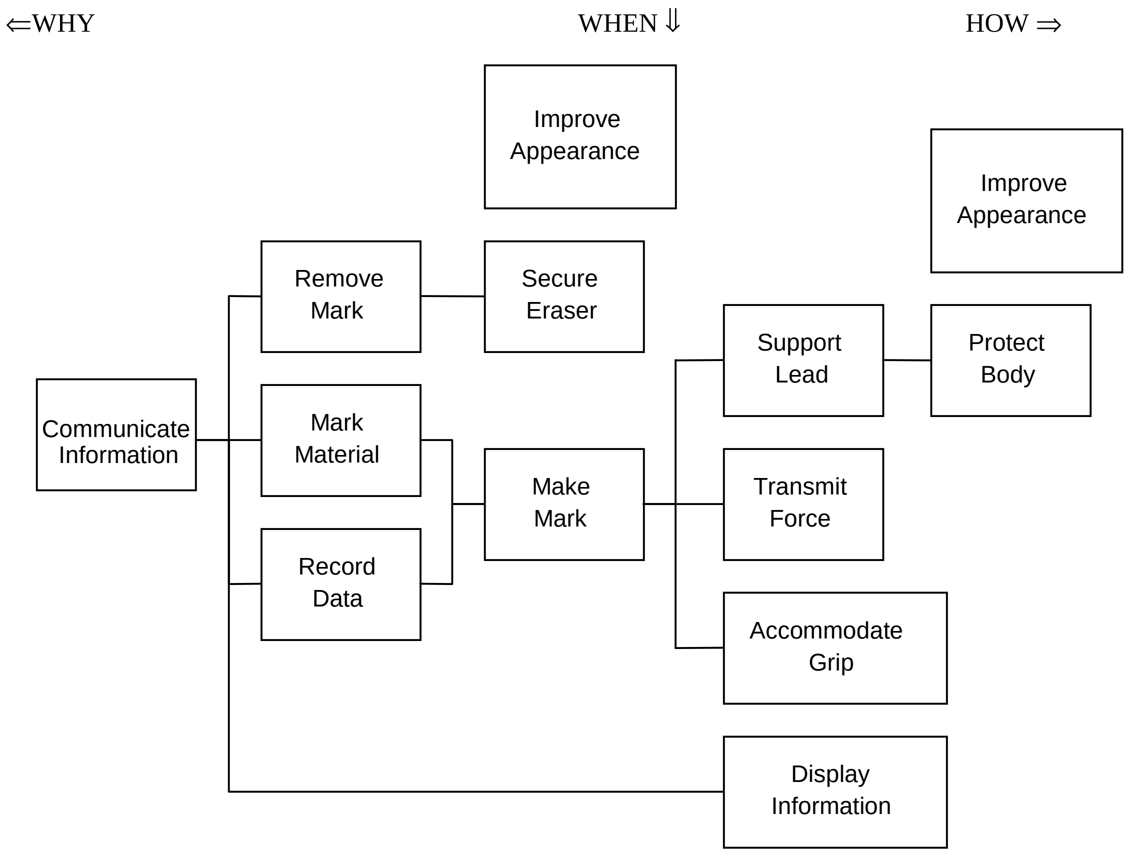
⇐WHY WHEN ⇓ HOW ⇒



A system can be complex or simple. Consider the following functions of the various components of a standard pencil.

COMPONENT	**FUNCTION**
Pencil (Assembly)	Communicate Information
	Mark Material
	Record Data
Body (Barrel)	Support Lead
	Transmit Force
	Accommodate Grip
	Display Information
Paint	Protect Body
	Improve Appearance
Lead (Graphite)	Make Mark
Eraser	Remove Mark
Band	Secure Eraser
	Improve Appearance

In tabular or list form these functions appear complete and are easily understood. Constructing a FAST diagram of these functions results in:



This diagram leads to questions that are not obvious from the list of functions, such as:

With the higher-order function of “Communicate Information,” the [potentially] least costly way to meet the function may be verbal communication. If so, is the pencil needed at all? If the pencil is needed, is the higher-order function really “Communicate Information,” or is it perhaps something like “Create Records?”

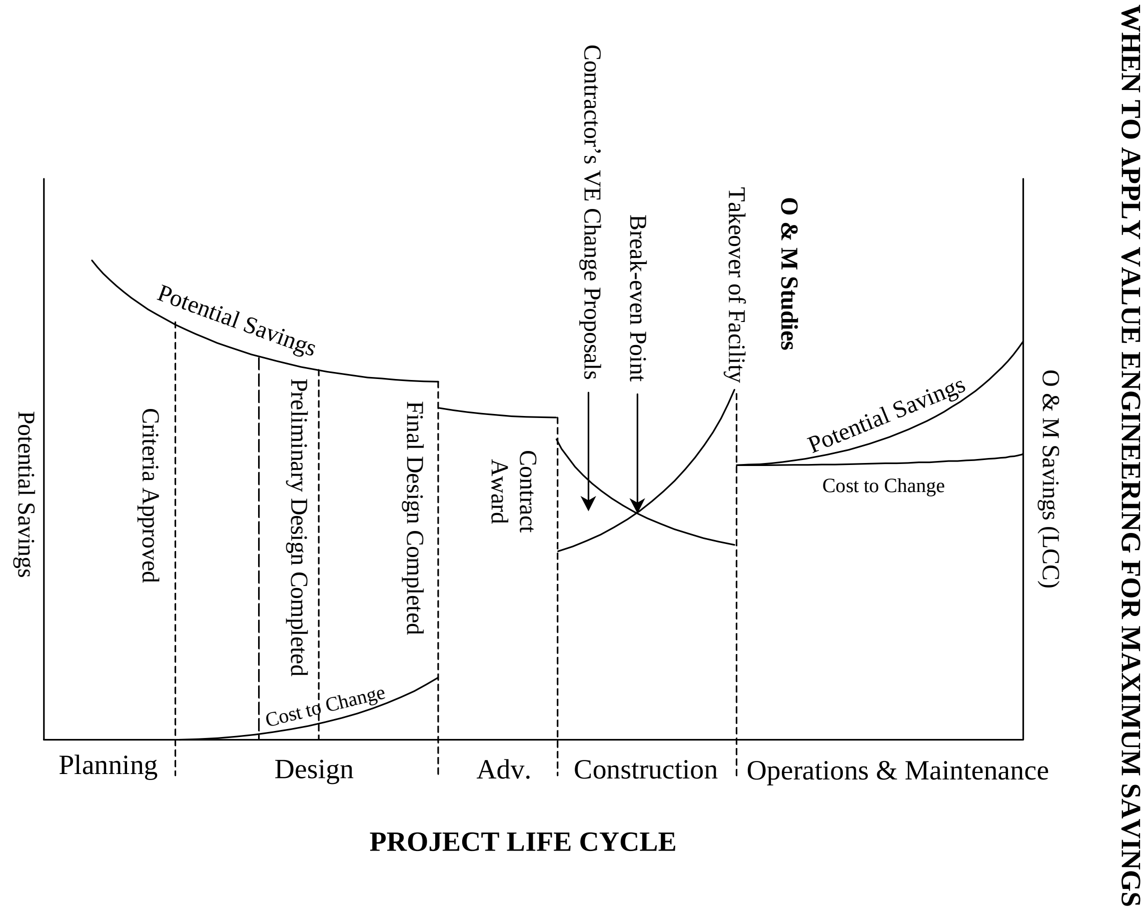
How does “Remove Mark” support the higher order function to “Communicate Information?” Is the component supporting this function (eraser) needed? Is there a function missing between them (e.g., “Obliterate Errors?”).

Does “Improve Appearance” support the function of the pencil? Is this needed? Does it cost anything (or is it just a benefit of accomplishing another function)? Are we missing a customer-focused function that does require improvements in appearance?

Such questions, and the answers to them, are fundamental to value engineering in helping to evaluate if the design approach responds to the functional needs of the activity.

ATTACHMENT 3

POTENTIAL FOR SAVING FROM VALUE ENGINEERING STUDIES AT VARIOUS STAGES IN A PROJECT CYCLE



ATTACHMENT 4

HIERARCHICAL MODELING OF CRITERIA

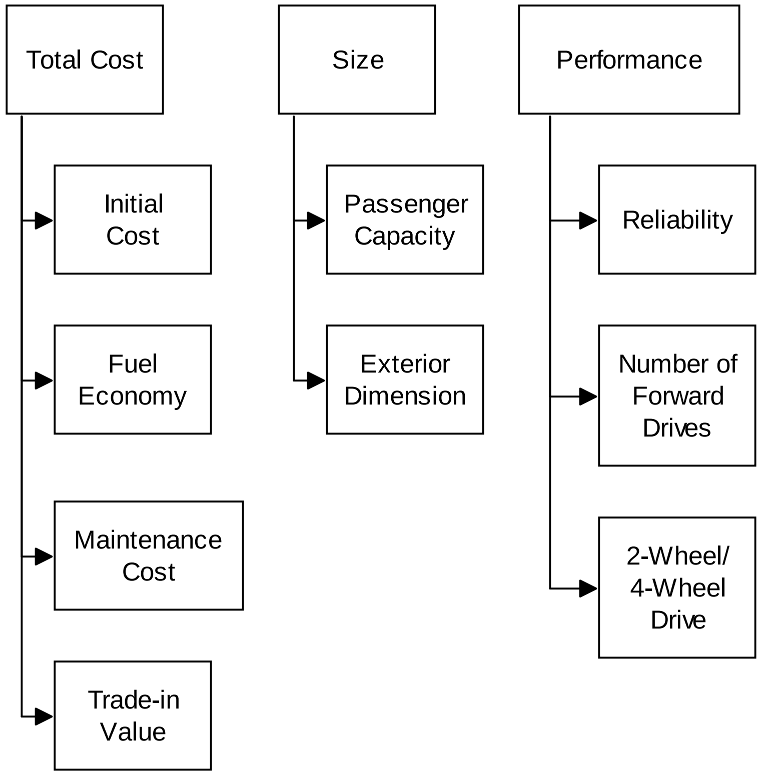
In many alternative studies there are a number of evaluation criteria identified that are not independent of each other or that are at such different levels of importance that direct comparison is difficult. In these cases it may be advantageous to group these dependent criteria into a structured hierarchical relationship. In a hierarchically structured criteria set, criteria are only evaluated against other criteria that are at the same level and under the same parent.

For example, suppose an objective is to buy new transportation. Without considering the specific alternatives, some criteria could be:

Total cost
Trade-in value
Maintenance cost
Performance
Fuel economy
Passenger capacity
Reliability
Exterior dimension
2-wheel/4-wheel drive
Number of forward drives/overdrive.

In comparing these criteria it would be very difficult to decide which is more important: total cost or maintenance cost, since maintenance cost is a part of total cost. It may be equally difficult to compare number of forward drives to total cost since they are such different levels that a direct comparison of which of these two is more important has little meaning.

If, however, this set of criteria is structured hierarchically, the “revised” criteria might appear as follows:



With the criteria in such a structure, only total cost, size, and performance are directly compared at the top level. At the next level, under the parent of total cost, initial cost, fuel economy, maintenance cost, and trade-in value would be compared relative to one another. By doing this, relative comparisons and relationships are easier to develop and understand.

ATTACHMENT 5

CRITERIA AND CRITERIA WEIGHTING

The table below illustrates a typical criteria weighting process. Each criteria is listed in both the row and column. Each set of criteria is then compared, once. The alpha in each block represents which of the two criteria being compared is the more important, while the number in each block represents by how much the dominant criterion is more important than the other. Once all comparisons are complete, the raw score for each criterion is determined by summing the numerical assigned to that alpha. These numbers are then either normalized to 10 (divide each score by the highest and multiply by 10) or converted to percents.

Criteria	A	B	C	D	E	
A. "Criterion A"		A1	A8	D2	A4	
B. "Criterion B"			B4	D3	B1	
C. "Criterion C"				D3	E2	
D. "Criterion D"					D1	
E. "Criterion E"						

Weighting Factors Legend:

No Difference — Very Important

1 2 3 4 5 6 7 8 9 10

Scores:

Criteria	A	B	C	D	E
Raw Score	13	5	0	9	2
Normalized Score	10	4	0	7	2
Percentage Score	.45	.17	--	.31	.07

ATTACHMENT 6

CRITERIA WEIGHTING IN THE ANALYTIC HIERARCHY PROCESS

The Analytic Hierarchy Process (AHP) uses matrix algebra and the eigenvector solution in an iterative process to determine criteria weights. An example of the process is as follows:[1]

1. Using pair-wise comparisons, an n^2 matrix is created where n is the number of criteria being compared. Values entered in the matrix are ratios of the importance, or priority, of one criterion over another. Values used in AHP generally range from 1 to 9, where 1 indicates equal importance and 9 represents almost an order of magnitude difference in importance.

   In the example shown below, criterion A is judged to be only half as important as criterion B (or $A_r/B_c = 1/2$), while criterion A is judged to be three times more important than criterion C (or $A_r/C_c = 3/1$). For the 3x3 matrix shown, the only remaining pair-wise comparison required is criterion B to criterion C, and in this example, criterion B is judged to be four times more important than criterion C (or $B_r/C_c = 4/1$). Since the diagonal of the matrix represents a comparison of each criterion against itself, each of these values, by definition, will be 1/1. The remaining matrix values (B_r/A_c, C_r/A_c, & C_r/B_c) are simply the reciprocals of the prior pair-wise comparisons.

	A_c	B_c	C_c
A_r	1/1	1/2	3/1
B_r	2/1	1/1	4/1
C_r	1/3	1/4	1/1

where: $_{r\ =\ row}$, and $_{c\ =\ column}$

2. The next step is to convert the fractional values to decimal equivalents of the desired precision,[2] and then compute the square of the matrix. For the example shown, $(A_r/A_c)^2 = (A_r/A_c \text{ x } A_r/A_c) + (A_r/B_c \text{ x } B_r/A_c) + (A_r/C_c \text{ x } C_r/A_c)$, or $(A_r/A_c)^2 = (1.0000 \text{ x } 1.0000) + (0.5000 \text{ x } 2.0000) + (3.0000 \text{ x } 0.3333) = 3.0000$. The remaining values of the squared matrix are calculated in a similar fashion.

	A_c	B_c	C_c				
A_r	1.0000	0.5000	3.0000		3.0000	1.7500	8.0000
B_r	2.0000	1.0000	4.0000	=	5.3332	3.0000	14.0000
C_r	0.3333	0.2500	1.0000		1.1666	0.6667	3.0000

[1] This example was extracted from the AHP Tutorial of the ECPro[TM] program CDROM available from Expert Choice, Inc., Pittsburgh, PA 15213.
[2] In this example, the desired level of precision is four decimal places.

3. Row sums are then calculated to produce the eigenvector solution and then normalized so that the sum is equal to 1. In the example below, criterion A has a value of 3.0000 + 1.7500 + 8.0000 = 12.7500, with a normalized value of 12.7500/39.9165 = 0.3194.

	A_c	B_c	C_c				
A_r	3.0000	1.7500	8.0000		12.7500		0.3194
B_r	5.3332	3.0000	14.0000	=	22.3332	=	0.5595
C_r	1.1666	0.6667	3.0000		4.8333		0.1211
			Total	=	39.9165		1.0000

(first iteration)

4. The process is then repeated using the calculated values from the matrix of the previous iteration until the difference between two consecutive solutions is less than a prescribed, or desired, value.[3] Using values from the solution of the previous matrix and squaring this new matrix yields the following results.

	A_c	B_c	C_c				
A_r	3.0000	1.7500	8.0000		27.6653	15.8330	72.4984
B_r	5.3332	3.0000	14.0000	=	48.3311	27.6662	126.6642
C_r	1.1666	0.6667	3.0000		10.5547	6.0414	27.6653

5. Row sums are again calculated to produce the eigenvector solution, and that result is then normalized.

	A_c	B_c	C_c				
A_r	27.6653	15.8330	72.4984		115.9967		0.3196
B_r	48.3311	27.6662	126.6642	=	202.6615	=	0.5584
C_r	10.5547	6.0414	27.6653		44.2614		0.1220
			Total	=	362.9196		1.0000

(second iteration)

6. The difference between the first two consecutive iterations is shown below. Since there is a difference to the fourth decimal place, an additional iteration is required.

	First iteration results		Second iteration results		Difference
A	0.3194	-	0.3196	=	-0.0002
B	0.5595	-	0.5584	=	+0.0011
C	0.1211	-	0.1220	=	-0.0009

[3] If the result of the iteration shows no change in the normalized value to the fourth decimal place, then another iteration is unnecessary.

7. Performing another iteration using the solution from the previous matrix and squaring this new matrix yields the following results.

	A_c	B_c	C_c				
A_r	27.6653	15.8330	72.4984		2295.7940	1314.0554	6016.8543
B_r	48.3311	27.6662	126.6642	=	4011.1349	2295.8740	10512.4476
C_r	10.5547	6.0414	27.6653		875.9853	501.3923	2295.7968

8. Row sums are again calculated to produce the eigenvector solution, and that result is then normalized.

	A_c	B_c	C_c					
A_r	2295.7940	1314.0554	6016.8543		9626.7037		0.3196	
B_r	4011.1349	2295.8740	10512.4476	=	16819.4565	=	0.5584	(third iteration)
C_r	875.9853	501.3923	2295.7968		3673.1744		0.1220	
			Total	=	30119.3346		1.0000	

9. The difference between the last two consecutive iterations is shown below.

	Second iteration results		Third iteration results		Difference
A	0.3196	-	0.3196	=	0.0000
B	0.5584	-	0.5584	=	0.0000
C	0.1220	-	0.1220	=	0.0000

10. Since there is no difference to the fourth decimal place, no additional iterations are required, and the criteria weights are defined by the values of the final iteration. For this example, the criteria weights are:

A	=	0.3196
B	=	0.5584
C	=	0.1220

For a more rigorous treatment of the complete Analytic Hierarchy Process, readers are referred to *Multicriteria Decision Making: The Analytic Hierarchy Process*, Vol. 1, AHP Series, Thomas L. Saaty, RWS Publications, Pittsburgh, PA – 1990, extended edition.

ATTACHMENT 7

ALTERNATIVE SELECTION

The table below illustrates a typical alternative selection process. The top number in each alternative score block represents the alternative's relative score against the identified criterion. The lower number represents the alternative's weighted score (Relative Score x Criterion Weight). Note that from this table alone it is not evident why alternative 1 received a relative score of 4 and alternative 2 a relative score of 3, against criterion A, etc.

This example indicates that alternative 3 is the preferred alternative.

CRITERION -------->	A	B	C	D	E				TOTAL SCORE
CRITERION WEIGHT --->	10	5	0	7	2				---
ALTERNATIVE	ALTERNATIVE SCORE/WEIGHTED SCORE								
1. "Alternative 1"	4/ 40	2/ 10		4/ 28	2/ 4				82
2. "Alternative 2"	3/ 30	3/ 15		4/ 28	3/ 6				79
3. "Alternative 3"	4/ 40	4/ 20		4/ 28	5/ 10				98
Note that in this example Criterion D did not contribute to the differentiation among the alternatives and could be eliminated.									

Alternative Scoring Legend:

Worst Choice Best Choice

1 2 3 4 5

ATTACHMENT 8

EXAMPLE MAUT ALTERNATIVE EVALUATION

The following simple MAUT example is for a common decision that most people have made, the decision of which vehicle to purchase. In this case the vehicles under consideration are sport utility vehicles (SUVs). The alternatives have been narrowed to three: A, B, and C. There are five decision criteria defined and weighted as follows:

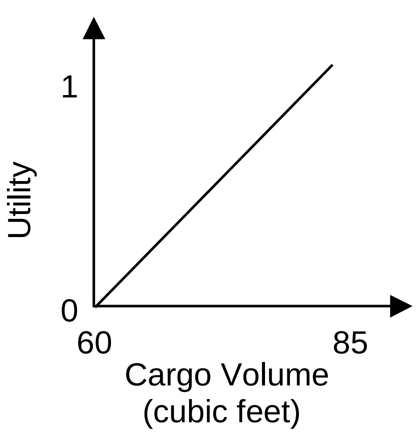
Weight	Decision Criteria
16%	maximize cargo volume
19%	maximize fuel mileage
24%	maximize horsepower
32%	minimize price
9%	maximize overall style and appearance.

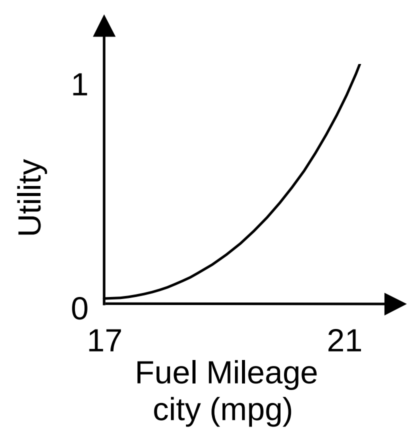
The alternative's performance against the decision criteria is given below in the Alternative Raw Values Table.

	Cargo Volume (cubic feet)	Fuel Mileage (mpg)	Horsepower	Price ($ x 1000)	Style/Appearance
Alternative A	85	17	210	32	Most Attractive
Alternative B	60	21	140	25	Least Attractive
Alternative C	78	18	173	28	Attractive

Alternative Raw Values

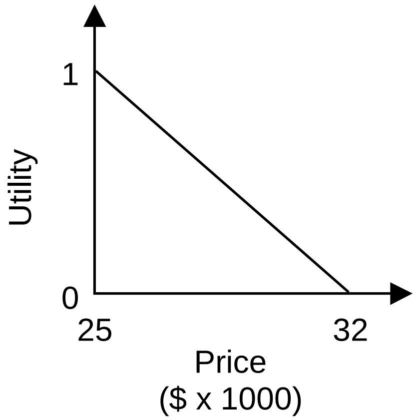
The decision maker generated the utility functions shown below for the decision criteria.

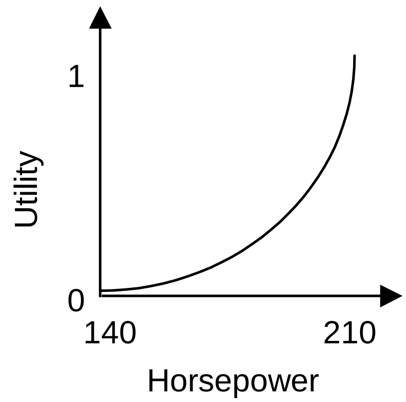




Style/Appearance Utility

Most Attractive	1.0
Attractive	0.5
Least Attractive	0.0





Given the attribute performance as provided in the Alternative Raw Value Table and the utility functions pictured above, the alternative utility scores can be determined. The Alternative utility scores can be found in the Alternative Utility Scores Table.

	Cargo Volume (cubic feet)	Fuel Mileage (mpg)	Horsepower	Price ($ x 1000)	Style/Appearance
Alternative A	1.00	0.00	1.00	0.00	1.00
Alternative B	0.00	1.00	0.00	1.00	0.00
Alternative C	0.72	0.09	0.33	0.57	0.50

Alternative Utility Scores

Now that the alternative utility values have been generated, the alternative ranking can be calculated in the same general manner as the example in Attachment 9. The overall ranking of the alternatives is calculated in the Alternative Ranking Calculation Table below. The first and second columns of the table provide the decision criteria and the criteria weights, respectively. The alternatives are listed across the top of the table in the first row. The alternative utility scores are repeated in the upper left-hand corner of the separated entries in the table. The weighted alternative utility scores are found in the lower right-hand corner of the separated entries of the table. The weighted utility scores are calculated by multiplying the utility score by the decision criteria weight. The weighted utility scores are then totaled to calculate an alternatives overall score.

As seen in the Alternative Ranking Calculation Table, the alternatives overall rankings are as follows:

ALTERNATIVE	OVERALL RANKING
Alternative A	0.49
Alternative B	0.51
Alternative C	0.44

This example results in an overall ranking with the alternatives scoring too close to make a decision. This decision analysis should not be completed at this point. Rather, a sensitivity analysis should be performed and the decision criteria should be reviewed for additional criteria that may further distinguish between the alternatives.

	Relative Weight	Alternative A		Alternative B		Alternative C	
Style/ Appearance	0.09	1.00	0.09	0.00	0.00	0.50	0.05
Cargo Volume	0.16	1.00	0.16	0.00	0.00	0.72	0.12
Horsepower	0.24	1.00	0.24	0.00	0.00	0.33	0.08
Fuel Mileage	0.19	0.00	0.00	1.00	0.19	0.09	0.02
Price	0.32	0.00	0.00	1.00	0.32	0.57	0.18
	Total		0.49		0.51		0.44

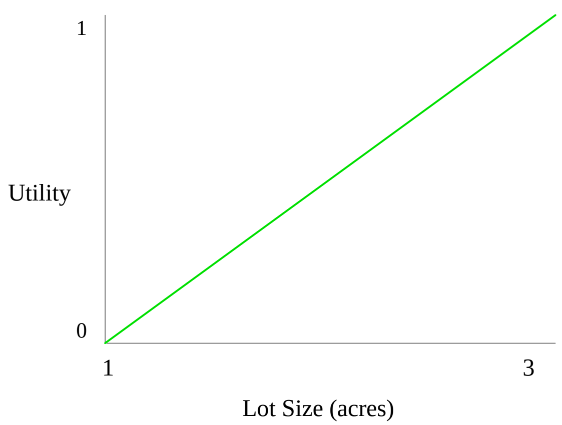
Alternative Ranking Calculation

ATTACHMENT 9

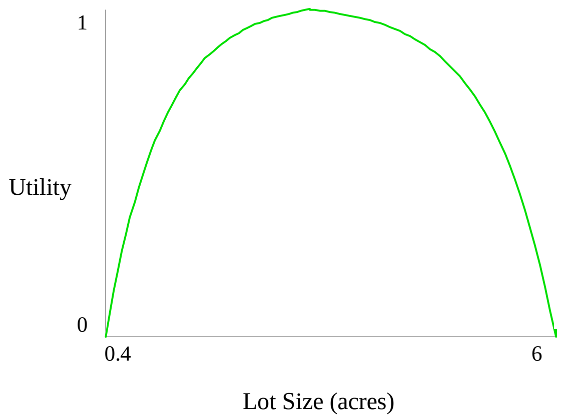
EXAMPLE UTILITY FUNCTIONS AND METHODS FOR GENERATING UTILITY FUNCTIONS

The following presents some of the possible utility functions that may be utilized to describe decision maker preferences. In these examples the decision criterion is related to the lot size and the decision being made is which home to purchase. There is a short discussion provided for each of the example utility functions in order to provide an idea of when the utility function may be applied.

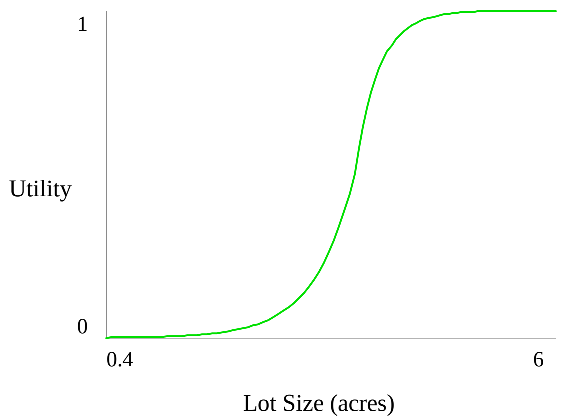
The straight-line utility function shown below is typically used when the range of performance for the feasible alternatives is reasonably close and there is no overwhelming preference for one end of the range over another. In this example, the prospective homeowner may have been interested in a home with a lot size of about 3 acres. The alternative homes had a relatively narrow lot size range of 1 to 3 acres and this resulted in a straight line utility function.



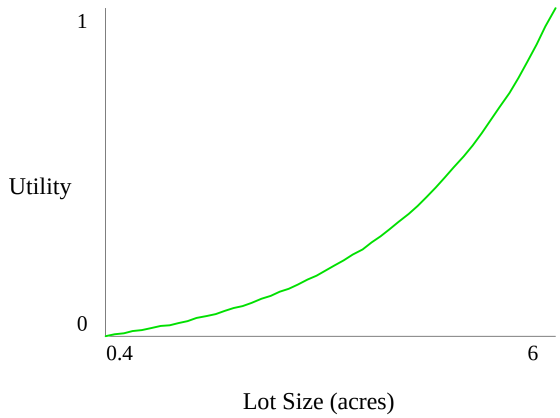
Like the example above, the prospective homeowner was interested in a home with a lot size of about 3 acres. In this case however, the alternative homes had lot sizes in a substantially larger the range from 0.4 to 6 acres. The decision maker may have felt that lots toward the smaller end of the range didn't offer adequate separation from the neighbors. Whereas the homes at the other end of the lot size range would involve too much yard work and therefore would be equally undesirable. With this utility function (shown below), homes with lots near the 3-acre point resulted in a higher utility score with respect to the lot size criterion.



This example utility function shown next again involves alternative homes with a lot size in the range from 0.4 to 6 acres. This utility function indicates that the decision maker values a home with a lot size in excess of 3 acres. Below 3 acres, the homes will receive a utility score close to 0. Above 3 acres, the homes will receive a utility score close to 1. Perhaps the decision maker in this case required a minimum of 3 acres to support animals and there was no aversion to a larger, 6-acre lot. This utility function closely resembles a go / no go requirement. In this example however, the homes with the lots less than 3 acres would have been eliminated had there been a requirement for lots with a minimum of 3 acres. Including a utility function similar to this in a decision analysis allows for the possibility of a home to be ranked high or the highest in the analysis because it performs very well with respect to other decision criteria rather than automatically be eliminated because of a requirement.



The shape of the next example utility function is the most common. In this example it is easily seen that the decision maker values the alternatives that have a larger lot size. The utility score remains relatively small until the lot sizes approach the larger end of the range when the utility scores increase rapidly.



The following describes some methods for generating utility functions that are supported by the Logical Decisions software. Details of the formulas and mathematical manipulations required to generate the utility functions are not provided, instead the choices and questions the decision maker must make are described. Additional information regarding the mathematics that the Logical Decisions® software employs to generate the utility functions can be found in the Logical Decisions for Windows™ decision Support Software User's Manual.

STRAIGHT LINE

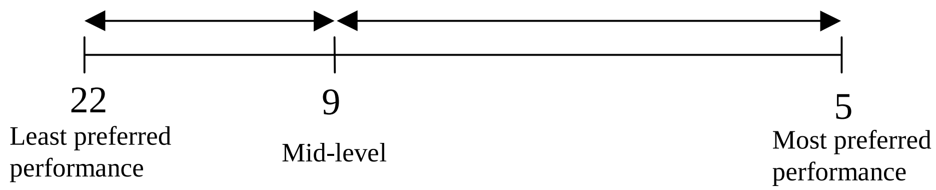
The most common type of utility function used is the straight-line utility function. To generate a linear utility function, typically the least preferred performance of the alternatives range of performance is assigned a utility of 0 and the most preferred level of performance is assigned a utility of 1. The utility function is then a straight line between the two points.

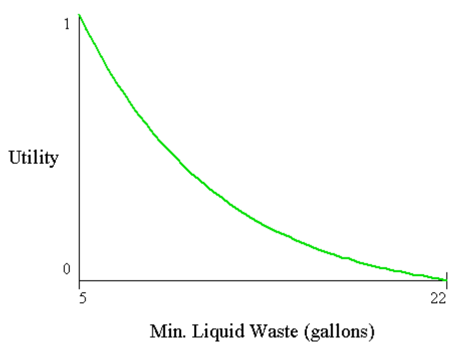
MID-LEVEL SPLITTING TECHNIQUE

This utility function generating technique seeks to establish the level of preference that is mid way between the least preferred and most preferred levels. The mid-preference level is identified by establishing two changes in the alternative performance level that have equal utility to the decision-maker. The figure below illustrates this. In this case, the decision-maker prefers the change from point A to point B in the same amount as the change from point B to point C. This technique assigns equal utility to changes 1 and 2 in order to generate the utility function.

Once the mid-level point is established, that point is assigned a value of 0.5 (for a utility scale of 0 to 1) and the utility function is drawn between the mid-level point and the least and most preferred levels. The example used in Section A.2.3.3 for the minimize liquid waste criteria is summarized below. The alternative performance, the mid-level, and the corresponding utility function are each shown.

	Alternative A	Alternative B	Alternative C
Liquid Waste (gallons)	22	5	15





When using this technique to generate the utility function, the decision-maker must answer a series of questions about changes in performance until the mid-level can be established. For the minimize liquid waste criteria example, these questions could have started with: "Is the change from 22 to 13.5 (13.5 is the mid-point between 22 and 5) gallons more important or the change from 13.5 to 5 gallons?" The decision-maker would have answered with "13.5 to 5 gallons." Then the range would have been narrowed and another question asked: "Is the change from 22 to 9 gallons more important or the change from 9 to 5 gallons"? In this example the decision-maker then would answer that the change from 22 to 9 and the change from 9 to 5 gallons are equally important. Therefore 9 is the mid-level preference. This is a very simplified example, and in practice this method will take more probing to arrive at the mid-level.

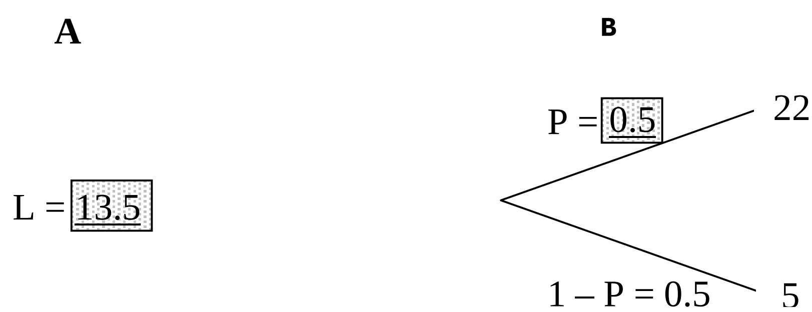
PROBABILITY TECHNIQUE

The probability technique allows the decision-maker to generate the utility function by answering a probability question. When this technique is employed, the decision-maker is asked to compare an alternative (A) that has a definite value for the decision criterion with another alternative (B) that has a lottery, or uncertain value, for the same decision criterion. Alternatives A and B differ only on the single decision criterion that the utility function is being generated for, they are equal with respect to the other criteria.

Consider the minimize liquid waste example above and the three alternatives A, B, and C. The alternative performance against the criteria is repeated below:

	Alternative A	Alternative B	Alternative C
Liquid Waste (gallons)	22	5	15

In this example the range of performance is between 5 gallons and 22 gallons and the mid-point is 13.5. The comparison in the Logical Decision software would start as:

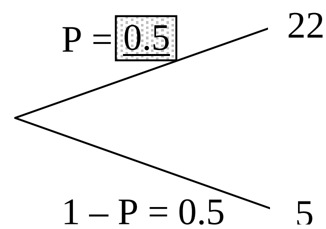


This default is asking the decision maker if a certain value of 13.5 gallons of waste produced is equal to a lottery with equal chance (P=0.5 and 1-P=0.5 or 50% probability) of ending up with 22 gallons or 5 gallons. If these two alternatives are equally preferable, the decision maker would indicate that and the utility function would be a straight line. More than likely, the default will not be equally preferable and the decision maker will be asked to adjust the certain outcome “L” and the probability “P” such that alternatives A and B are equally preferable.

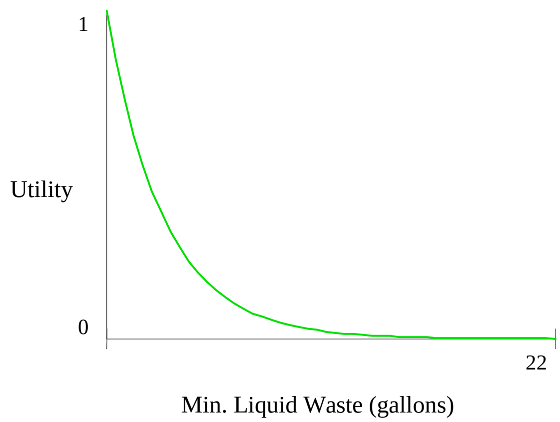
Assuming that the decision maker adjusts “L” to 6.5 and indicates that alternatives A and B are equally preferable, the utility function shown after the equal alternatives A and B would be generated.

A

B



L = 6.5



ATTACHMENT 10

TYPICAL ALTERNATIVE STUDY REPORT CONTENTS

Abstract or Forward

Introduction

Provide a general description of the scope, purpose, and timing of the study.

Background

Provide a brief description of the activity being studied.

List of Participants

Identify the study participants.

Study Limitations and Assumptions

Identify any limitations imposed on the study and any key assumptions.

Methodology

Describe the methodology used in the conduct of the study.

Discussion of Results

Provide a detailed discussion of the evaluation(s) conducted and the results of the evaluation(s).

Summary/Conclusions

Provide a summary of the results of the study.

Recommendations

Identify recommendations resulting from the study.

Attachments

For value engineering studies, the FAST diagram is included, either in the methodology, results, or attachments.